

INHALTSVERZEICHNIS

1.	Einleitung	1
2.	Lastkontrolle.....	5
2.1.	Der Aufbau eines Transaktionssystems	5
2.2.	Modellkonzeption	7
2.3.	Aufgaben der globalen und lokalen Lastkontrolle	9
3.	Lokale Lastkontrolle	11
4.	Das Simulationssystem	20
4.1.	Aufbau des Systems	20
4.2.	Modellierung der Last	24
4.3.	Der Transaktionsverwalter	25
4.4.	Der Regelungskreislauf	28
4.4.1	Bewertung der Lastsituation.....	29
4.4.2	Ursachenbestimmung.....	31
4.4.3	Regelung des Transaktionssystems	36
4.5.	Modifikationen des Simulationssystems	40
4.5.1	Veränderungen in der topology-Datei.....	40
4.5.2	Veränderungen im Lastkontrollmodul	41
4.5.3	Veränderungen im CC-Modul.....	41
5.	Lastkontrolle bei CPU-Engpässen.....	43
5.1.	Simulationen mit stabiler Last	46
5.1.1	Variation der Ankunftsraten und des MPL-Startwertes	46
5.1.2	Antwortzeit-Aufgliederung bei statischer MPL-Einstellung... ..	47
5.1.3	Ergebnisse der globalen MPL-Regelung.....	48
5.1.4	Variation des MPL- Startwertes für zwei Lastsituationen	52
5.2.	Simulationen mit variabler Last	55
5.3.	Sensitivitätsanalyse	57
5.3.1	Variation der Antwortzeitanteile.....	57
5.3.2	Regelung über den CPU-Auslastungsindikator.....	61
5.4.	Simulationen mit 2 Transaktionstypen	63
5.5.	MPL-Regelung und Prioritätsalgorithmen	70
5.5.1	Ergebnisse bei statischer MPL-Einstellung.....	71
5.5.2	Ergebnisse beim GMPL-Verfahren.....	72
5.5.3	Ergebnisse bei unterschiedlicher Verwendung von Prioritäten	73

**Simulative Bewertung
von
zielorientierten Lastkontrollverfahren**

Diplomarbeit von Ralf Wagner

Universität Leipzig
Institut für Informatik
Abteilung Datenbanken
Prof. Dr. habil. E. Rahm

Aufgabenstellung und Betreuung : Prof. Dr. habil. E. Rahm

September 1995

6. Lastkontrolle bei Sperrengpässen..... 76

6.1. Simulationen mit einem Transaktionstyp78

6.1.1 Bestimmung der Szenarien..... 80

6.1.2 Bestimmung des optimalen statischen MPL-Wertes..... 85

6.1.3 Sensitivitätsanalyse..... 90

6.2. Simulationen mit zwei Transaktionstypen97

6.2.1 Simulationen mit konstantem Ankunftsverhalten 97

6.2.2 Simulationen mit stark variierender Last..... 110

6.3. Simulationen mit einer realen Last119

6.3.1 Bestimmung eines optimalen statischen MPL-Wortes 121

6.3.2 Ergebnisse der globalen MPL-Regelung 125

6.3.3 Ergebnisse der lokalen MPL-Regelung..... 129

7. Zusammenfassung und Ausblick 131

8. Anhang..... 135

8.1. Literaturverzeichnis135

8.2. Anhang A137

8.3. Anhang B138

8.4. Versicherung143

1. Einleitung

Durch den breiten Einsatz von Datenverarbeitungssystemen haben sich in den vergangenen Jahren sowohl der Umgang mit Informationen als auch die Informationsbedürfnisse gewandelt. Auch für den Bereich der Datenbanksysteme ergeben sich hierdurch zusätzliche Anforderungen. Diese Anforderungen äußern sich im Betrieb durch eine weitaus größere Heterogenität der zu bewältigenden Arbeitslast. Dieser Trend wird sich durch eine Reihe von Faktoren noch weiter fortsetzen. Zu nennen sind hierbei unter anderem (vgl. [MHMZ94]):

- (a) Die Lasten werden immer komplexer: Online Transaction Processing (OLTP) wird kombiniert mit Anfragen für Management-Informationssysteme (MIS)
- (b) Die Verfügbarkeit von leistungsstarken Applikationsgeneratoren führt zur Erweiterung des Programmkatalogs um Aufgaben, die vormalig aus Kostengründen zurückgestellt wurden.
- (c) Die Integration verschiedenster Hardwareplattformen und der Einführung von Daten-Transparenz aus Applikationssicht führen zu einem kaum mehr überschaubaren Arbeitsaufkommen.

Um diesem Trend von der technischen Seite Rechenschaft zu leisten, ist es notwendig, die vorhandenen Hardwarekapazitäten möglichst gut auszunutzen. Für den Bereich der Datenbanken bedeutet dies eine den jeweiligen Verhältnissen angepaßte optimale Systemverwaltung. Eines der Hauptprobleme heutiger OLTP-Systeme ist jedoch die Komplexität dieser Verwaltungsaufgaben. Diese Aufgaben können in der Regel nur von einigen Spezialisten wahrgenommen werden. Deren Aufgabe untergliedern sich hierbei in die eng miteinander verbundenen Bereiche der Lastkontrolle, der Betriebsmittelvergabe und der Leistungsoptimierung. Generell läßt sich die augenblickliche Situation wie folgt beschreiben (vgl. [RFL89, BCL93]):

- Die Systemverwaltung wird dominiert durch eine Vielzahl von Low-Level-Schnittstellen, die schwierig zu verstehen und zu nutzen sind. Für jede Gruppe von Arbeitslasten müssen eine große Anzahl von internen Parametern (Prioritäten, Speicherbedarf etc.) zur Steuerung der Betriebsmittelvergabe festgesetzt werden. Desweiteren gibt es Steuerungsparameter für diejenigen Subsysteme (Data Base Management System (DBMS), Transaction Processing-Monitor), die auf der Ebene oberhalb des Betriebssystems laufen.

- Falls die Last auf verschiedene Verarbeitungsknoten aufgeteilt werden soll, wird die Komplexität noch vergrößert. Als zusätzliche Steuerungsprobleme treten dabei Fragen nach der Rechnerallokation von Subsystemen, Applikationsprogrammen und Transaktionen auf.
- Die Steuerung in heutigen Systemen wird zum größten Teil manuell abgewickelt. Dies bedeutet aber auch, daß eine flexible und vor allem schnelle Anpassung der lokalen und globalen Kontrollparameter nicht erfolgen kann. Die Erkennung von Leistungsproblemen und die Durchführung von Gegenmaßnahmen liegen somit in der Hand des Systemverwalters. Dies und die Tatsache, daß die meisten Analysen erst nach Entnahme eines Mitschnittes (Traces) der Last möglich werden, machen klar, daß zwischen dem Auftreten und der Behebung eines Problems eine geraume Zeit vergehen kann.

Es liegt auf der Hand, daß die Aufhebung dieser Beschränkungen eine sehr komplexe Aufgabe darstellt. Das Ziel solcher Bestrebungen ist ein automatischer, sich selbst regulierender Mechanismus für die Leistungskontrolle. Die Systemverwaltung soll soweit vereinfacht werden, daß die Steuerung direkt durch die Vorgabe von externen Zielgrößen geschehen kann, anstatt diese Ziele in interne, maschinenorientierte Parameter übersetzen zu müssen. Als **mögliche Zielgrößen** kommen dabei folgende Maße in Betracht [RFL89]:

(1) **Antwortzeiten**

Die mittleren Bearbeitungszeiten einer Lastgruppe sollen einen vorgegebenen Wert nicht überschreiten.

Diese Art von Vorgaben sind vor allem bei interaktiven Transaktionen zu treffen.

(2) **Durchsatz**

Die Vorgabe von Durchsatzzielen ist primär für den Stapelbetrieb von Interesse. Hier treten die Antwortzeiten in den Hintergrund. Aber auch für die Antwortzeiten ist der Durchsatz relevant, da im allgemeinen davon ausgegangen werden kann, daß die beiden Größen miteinander korreliert sind.

(3) **Deadlines**

Deadlines beziehen sich auf individuelle Transaktionen [Vo93], die zu einem bestimmten Zeitpunkt beendet sein sollen. Dies entspricht der Vorgabe einer maximal erlaubten Antwortzeit. Diese Art von Leistungs- bzw. Zielgrößen sind besonders wichtig für Realzeit-DBS [EN89].

(4) **Service-Rate**

Ist der Ressourcenbedarf für bestimmte Transaktionstypen nicht bekannt oder nicht absehbar, so kann es opportunistisch sein, einer Transaktion einen bestimmten Anteil an einzelnen Ressourcen zu zuweisen. So könnte z.B. einer Transaktionsklasse ein bestimmter Prozentsatz der CPU Kapazität zugewiesen werden.

Für den Entwurf des hier verfolgten Regelungsansatzes wurde die **Zielgröße Antwortzeit** gewählt, da sie sowohl eine wichtiges Charakteristikum in Bezug auf die Leistungsbewertung ist, als auch eine an der Praxis orientierte Größe verkörpert. Die Wahl der Antwortzeit als Zielgröße folgt somit dem Leitgedanken der Ausrichtung der Steuerungsparameter auf den Menschen. Dies soll auch Nicht-Fachleuten ermöglichen, ein Datenbanksystem (DBS) optimal zu nutzen.

Die hier verfolgte Konzeption stellt ein **zielorientiertes integriertes adaptives Lastkontrollverfahren** dar. Dieser ganzheitliche Ansatz weicht somit von vielen Untersuchungen dahingehend ab, daß diese zumeist nur einen einzelnen Engpaß betrachten (vgl. z.B. [AG89, CKL89, HLC91, PLC92, Th92]). Die Behandlung von Engpässen, die auf verschiedenen Ebenen auftreten können, werden hier nicht durch mehrere isoliert operierende Verfahren sondern durch ein einzelnes Verfahren vorgenommen, welches entsprechend der Lastsituation verschiedene Systemparameter anpaßt.

Die hier vorliegende Untersuchung baut auf der Diplomarbeit von G.S.Krink [Kr94] und dem Simulationssystem TPSIM [St91, St93, Ma91, Wo91] auf. Im Rahmen der Diplomarbeit von G.S. Krink wurde eine Verfahren zur zielorientierten adaptiven Lastkontrolle entwickelt und in ein bestehendes Simulationssystem integriert. Primäre Aufgabe der vorliegenden Diplomarbeit ist die **quantitative Bewertung** der implementierten Kontrollmethoden durch Simulationsexperimente.

Dominierendes Ziel des Systems ist die Einhaltung der vorgegebenen Antwortzeiten für die einzelnen Transaktionsklassen. Abgeleitet aus diesem Ziel ist die Frage nach der Stabilität des Systems bei verschiedenen Laststrukturen und der Wahl der internen Steuerungsparameter der Regelungskomponente. Ein weiterer wichtiger Aspekt ist auch die Sensitivität der Steuerungsparameter. Eine vollständige Kapselung der Parameter läßt sich nur dann erreichen, wenn sich diese als hinreichend robust für verschiedene Situationen erweisen.

Die Arbeit gliedert sich in folgende Teilbereiche: Das zweite und dritte Kapitel sind der generellen Beschreibung der Problemkomplexes sowie möglichen Lösungsmöglichkeiten gewidmet. Im vierten Kapitel werden das Simulationssystem und die Regelungskomponente

beschrieben. Im fünften Abschnitt werden verschiedene Simulationen, die das Verhalten des Systems bei einem CPU-Engpaß wiedergeben, beschrieben und erklärt. Im sechsten Kapitel wird danach auf die Sperrenpaßbehandlung eingegangen. Ebenfalls Bestandteil dieses Abschnittes sind Simulationen, die mit einem Mitschnitt aus einer realen Last erzeugt wurden. Im siebten Kapitel wird ein Resumé und ein Ausblick auf mögliche Erweiterungen und Verbesserungen gegeben.

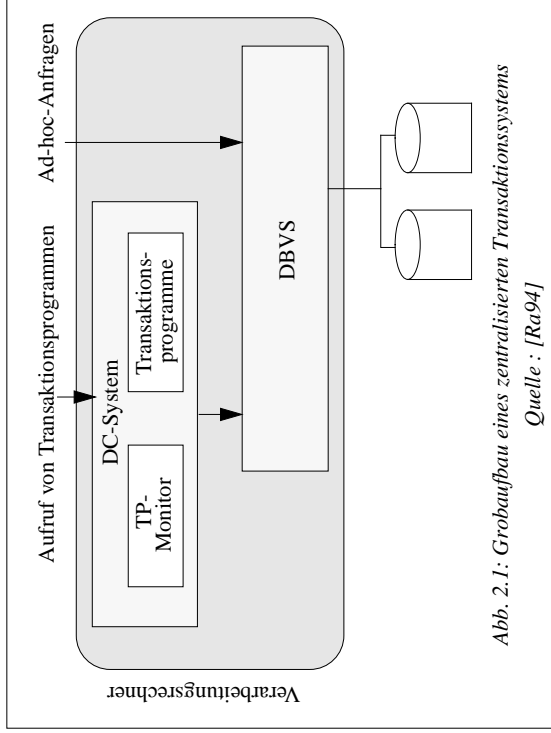
2. Lastkontrolle

Bevor im Kapitel 4 die einzelnen Module und die implementierten Methoden detailliert erläutert werden, ist es Aufgabe dieses Kapitels, die grundlegenden Überlegungen zum Thema der Laststeuerung darzulegen. Unter dieser Bezeichnung werden alle Funktionen, die mit dem Scheduling von Lasteinheiten (bzw. Allokation von Betriebsmitteln) sowie der Überwachung der Lastverarbeitung zusammenhängen, subsumiert (vgl. [RFL89]). Diesen Überlegungen vorangestellt wird eine Übersicht des Aufbaus eines Transaktionssystems und eine kurze Beschreibung des unterstellten DBS-Modells. Aus den für die Lastkontrolle definierten Aufgaben werden mögliche Vorgehensweisen abgeleitet. Diese Überlegungen umfassen unter anderem auch die Wahl der Parameter zur eigentlichen Ursachenerkennung.

2.1 Der Aufbau eines Transaktionssystems

Ein Datenbanksystem besteht aus der eigentlichen Datenbank (DB) sowie dem Datenbankverwaltungssystem (DBVS) bzw. Datenbank-Management-System (DBMS). Die Datenbank umfaßt die Gesamtheit der gesammelten Daten und ist im allgemeinen auf Externspeicherung permanent gespeichert. Das DBVS stellt die Zugriffsschnittstelle auf die Daten dar. Es verwaltet die Daten und führt sämtlicher Operationen bzw. Zugriffe auf sie aus. Ein Datenbankzugriff erfolgt innerhalb von Transaktionen, die aus einer oder mehreren DB-Operationen bestehen können.

DBS werden in der kommerziellen Datenverarbeitung vorwiegend im Rahmen von Transaktionssystemen eingesetzt. Diese Systeme bieten die Möglichkeit vorgeplante Anwendungsfunktionen (z.B. Überweisungen zwischen Konten, Lagerveränderungen) durch entsprechende Anwendungs- oder Transaktionsprogramme durchführen zu lassen (vgl. [Ra94, Hä+87]). Abb. 2.1 zeigt den Grobaufbau eines zentralisierten Transaktionssystems.



Ein Transaktionssystem besteht aus dem DBVS und einem sogenannten DC-System (Data Communication System). Das DC-System besteht aus dem TP-Monitor (Transaction Processing Monitor) und einer Menge von Transaktionsprogrammen.

Die Aufgabe der TP-Monitor bestehen in der Kontrolle der Ausführung der Transaktionsprogramme, der Kommunikation von Programmen mit Terminals sowie mit der Kommunikation mit dem DBVS. Zur Realisierung der Kommunikation wird zumeist auf die entsprechenden Funktionen des Betriebssystems zurückgegriffen. Aus Leistungsgründen wird die Prozeß- und Programmverwaltung meist durch den TP-Monitor selbst realisiert.

2.2 Modellkonzeption

Als Grundlage für die Gestaltung des Simulationsmodells dient ein lokal verteiltes Transaktionssystem. Technisch besteht ein solches System aus einer bestimmten Anzahl von Verarbeitungsknoten, die über ein Hochgeschwindigkeitsnetz miteinander verbunden sind. Die Last erreicht über ein Kommunikationsnetz das System und wird auf die einzelnen Knoten verteilt. Jeder einzelne Knoten ist mit vollständiger Funktionalität (d.h. mit Subsystemen, wie TP-Monitor, DBVS) sowie eigener CPU und eigenem Hauptspeicher ausgestattet. In jedem Knoten läuft somit eine Kopie des DBVS, die mit den anderen DBVS-Instanzen kooperiert. Die physische Zuordnung der Datenbestände kann entsprechend einer Shared-Disk- bzw. Shared-Nothing-Architektur (vgl. [Ra94]) erfolgen. Die Zugriffsmöglichkeit auf die Gesamtheit der Daten ist entsprechend an allen Knoten vorhanden (Shared-Disk) bzw. wurde zwischen den Knoten verteilt bzw. partitioniert (Shared-Nothing). Bestandteil jedes Subsystems ist ein Scheduler, der die Aufgabe der Zuweisung der Arbeitseinheiten zu Tasks oder Prozessen übernimmt. Zwischen der Lastkontroll-Komponente und den Monitor-Einheiten existieren Möglichkeiten zum Datenaustausch.

Vom konzeptionellen Standpunkt aus werden zwei Komponenten hinzugefügt. Dies geschieht zum einem auf der globalen Ebene und zum anderen auf der lokalen Ebene. Abb. 2.2 stellt die Grundzüge eines lokal verteilten Systems dar, das um die beiden Komponenten ergänzt wurde. Die Lastkontroll-Komponenten benötigen zur Steuerung Informationen über den Systemzustand. Diese erhalten sie von den Monitor-Komponenten, welche im Betriebssystem, im Datenbanksystem oder im TP-Monitor angesiedelt sein können. Die globale Lastkontrolle erhält weitere Daten von den lokalen Lastkontroll-Komponenten.

isolierter Kontrollansatz, widerspricht jedoch dem Ziel der Integration der Steuerung verschiedener Subsysteme. Zudem erschwert eine verteilte Anordnung die Koordinierung. Es bietet sich an, die Steuerung in einer zyklisch zu durchlaufenden Rückkopplungsschleife vorzunehmen.

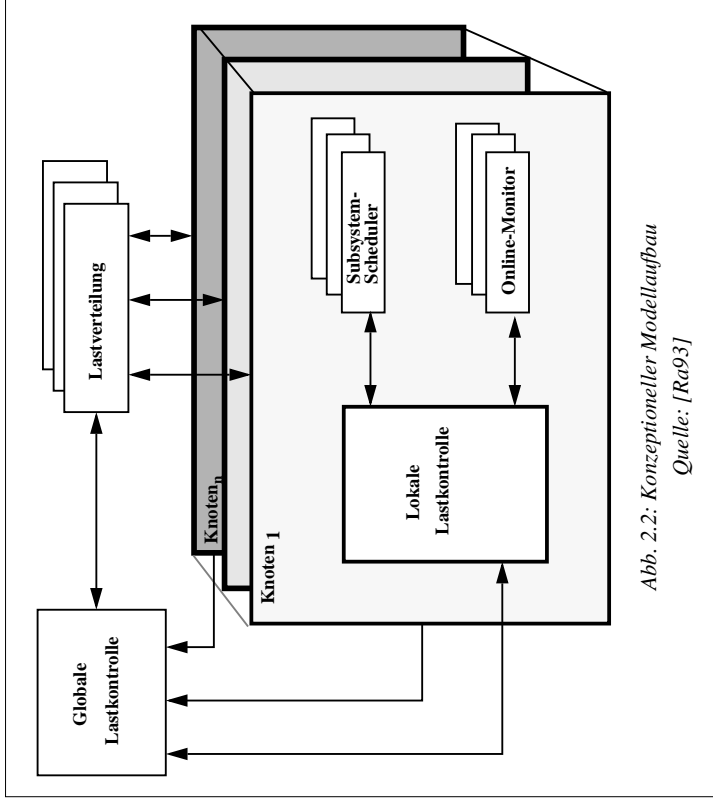


Abb. 2.2: Konzeptioneller Modellaufbau

Quelle: [Ra93]

Die Funktionsweise der beiden Lastkontrollkomponenten sind in Abb. 2.3 abgebildet. Die eigentliche Datensammlung findet in den Monitor-Komponenten statt. Die Lastkontrolle greift auf die dort verfügbaren Daten zu und evaluiert diese im Hinblick auf mögliche Ausnahmesituationen bzw. Leistungsprobleme. Wird ein Problem bzgl. der Zielgrößen festgestellt, so nimmt die jeweilige Lastkontrollkomponente unter Rückgriff auf die Monitor-Daten eine Ursachenbestimmung vor. Nach der Ursachenbestimmung werden korrigierende Maßnahmen (tuning) zur Problembeseitigung ergriffen. Dies geschieht, indem die Steuerungsparameter der Scheduler-Komponenten verändert werden.

Die hier gewählte Darstellung der Lastkontrolle als eigenständiges Modul ist lediglich aus konzeptionellen Gründen gewählt. Die Realisation einer solchen Kontroll- bzw. Steuerungsinstanz ist rein theoretisch auch integriert in die bestehenden Komponenten denkbar. Ein solcher

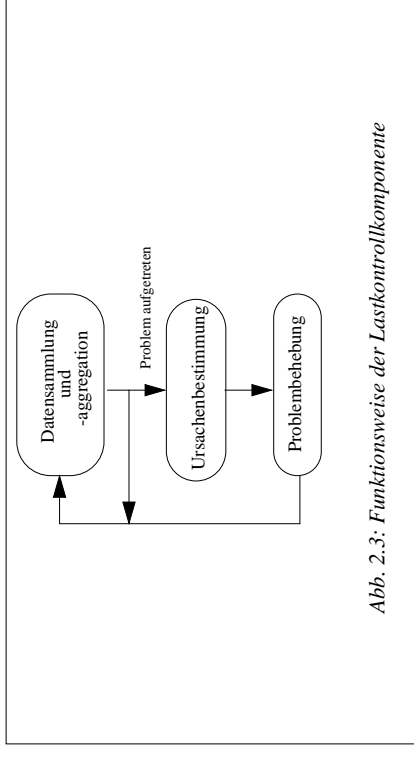


Abb. 2.3: Funktionsweise der Lastkontrollkomponente

2.3 Aufgaben der globalen und lokalen Lastkontrolle

Obwohl in der vorliegenden Arbeit grundsätzlich vom Ein-Rechner-Fall bzw. von einem Verarbeitungsknoten ausgegangen wird, wird der Vollständigkeit halber noch kurz auf die globale Lastkontrolle eingegangen.

Die Aufgaben der **globalen Lastkontrolle** können in vier Kategorien unterteilt werden:

- (1) Zuweisung der Transaktionen zu den einzelnen Verarbeitungsknoten
- (2) Bereitstellung einer einheitlichen Verwaltungsschnittstelle
- (3) Sammeln und Auswerten von systemspezifischen Daten für die Kontrollentscheidungen
- (4) Reaktion auf Problemzustände (z.B. Anpassung der Verteilungsstrategie)

Die Hauptaufgabe der globalen Lastkontrollkomponente liegt in der Lastbalancierung und der globalen Zielgrößenkontrolle (Punkte 1/3/4). Beide Funktionen können sowohl auf Front-End-Knoten als auch auf den regulären Rechnerknoten angesiedelt werden. Die globale Lastkon-

trolle greift in die Verteilung der Arbeitslast auf die Rechnerknoten ein, um zum einen die Überlastung von Knoten zu vermeiden und zum anderen, um eine optimalen Leistung zu erreichen. Da von einem lokal gekoppelten System ausgegangen wird, ist die Verteilung der Last unter Berücksichtigung der Referenzlokalität zur Minimierung des E/A- und Kommunikationsaufwandes ein wesentlicher Faktor zur optimalen Justierung des Systems (vgl. [Ra89, FNG93]).

Die Aufgabe der **lokalen Lastkontrollkomponente** ist die Überwachung der zugewiesenen Arbeit auf einem spezifischen Verarbeitungsknoten. Auch hier ist das Globalziel die Einhaltung vorgegebener Antwortzeitschranken. Untersucht werden an dieser Stelle einzelne Systembereiche, wie z.B. CPU oder E/A. Beim Auftreten eines Problems ist es die Aufgabe der lokalen Kontrolle den jeweiligen Verursacher bzw. die Engpaßkomponente zu bestimmen und korrigierende Maßnahmen vorzunehmen. Im folgenden Kapitel behandelt den von der lokalen Lastkontrolle zu bewältigen Aufgaben genauer.

3. Lokale Lastkontrolle

Die Aufgabe der lokalen Lastkontrollelemente ist die Ressourcenallokation an den einzelnen Knoten. Diese Ressourcen lassen sich in die Gruppe der **physischen** und **logischen** Ressourcen unterteilen. Zu der Gruppe der physischen Ressourcen gehören die CPU und der Hauptspeicher, die in der Regel vom Betriebssystem kontrolliert bzw. gesteuert werden. Die logischen Ressourcen, wie z.B. die Sperren, werden in den einzelnen Subsystemen gesteuert.

Eine Form von logischen Ressourcen bilden die Server-Komponenten wie das DBMS und der TP-Monitor. Sie stellen bestimmte Dienste für verschiedene oder gleiche Anforderungstypen bereit. So verarbeitet das DBMS Datenbankanfragen, die ihm von den Ausführungsprogrammen übergeben werden; der TP-Monitor stellt Dienste zur Kommunikation zur Verfügung. Auf diese Weise kann das ausführende Programm von den hardware-spezifischen Eigenschaften der verschiedenen Kommunikationseinheiten isoliert werden.

Ein typisches Beispiel für logische Ressourcen auf der Ebene des TP-Monitors sind die **Ausführungseinheiten**. Die Steuerung dieser Ressource wird über die Vorgabe der Anzahl maximal aktiver Ausführungseinheiten vorgenommen. Dieser Wert wird in der Literatur als **Multiprogramming Level (MPL)** bezeichnet. In verschiedenen Systemen existiert auch die Möglichkeit, den statischen globalen MPL-Wert weiter spezifizieren zu können. Es können für einzelne Transaktionsklassen individuelle statische MPL-Werte eingestellt werden.

Eine zu hohe Wahl dieses Parameters kann durch zu hohe CPU-Belastung oder zu große Sperrkonfliktrate zu negativen Auswirkungen auf die Antwortzeit oder den Durchsatz führen. Eine niedrige Wahl dieses Wertes führt jedoch zu einer schlechten Auslastung der physischen Ressourcen. In momentan verfügbaren DB-Systemen wird der MPL-Wert als konstante Größe festgesetzt. In der Regel kann man davon ausgehen, daß dieser Wert über einen relativ langen Zeitraum unverändert bleibt. Das Transaktions-Scheduling innerhalb des DB/DC-Systems unterliegt weiteren Systemparametern. Diese bestimmen u. a. die Transaktionsprogramme für bestimmte Ausführungseinheiten, wie viele Transaktionen einer bestimmter Klasse zur selben Zeit bearbeitet werden dürfen oder welche Prioritäten bestimmten Transaktionsklassen oder Transaktionen zugeordnet werden. Die Einstellung oder Modifikation dieser Parameter erfolgt wiederum manuell durch die Systemverwalter.

Die Möglichkeiten der Einführung einer automatischen lokalen Lastkontrolle werden in bestehenden Systemen durch die Trennung der einzelnen eigenständigen Komponenten wie Betriebssystem, DBMS etc. erschwert. Im folgenden werden die einzelnen Bestandteile des Systems auf die wesentlichen Aspekte reduziert. Im Modell besteht das System aus Prozessoren, E/A-Geräten, einem TP-Monitor und einer Sperrkomponente. Andere Komponenten wie die Pufferverwaltung oder das Caching des Betriebssystems sind hier nicht Gegenstand der lokalen Lastkontrolle. Bei Konzentration auf die wesentliche Teilmenge lassen sich die Ressourcen entsprechend Abb. 3.1 gliedern.

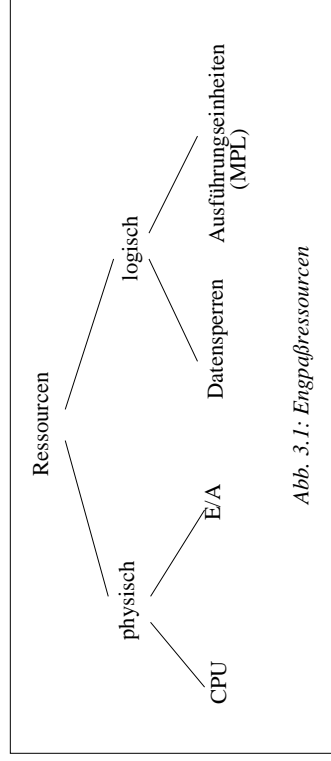


Abb. 3.1: Engpaßressourcen

Die einzelnen Komponenten aus Abb. 3.1 stellen auch zugleich die möglichen Engpässe dar. Um die maximale Leistungsfähigkeit des Systems zu erreichen, sollte jede dieser Komponenten bis zum Sättigungspunkt belastet werden können [Th92]. Die Sättigungsgrenze (**point of diminishing returns, thrashing point**) einer Engpaßressource wird erreicht, wenn sich jede weitere Belastung negativ auf die Zielvorgaben, also hier die Antwortzeit, auswirkt. Die Steuerung muß jedoch verhindern, daß dieser Punkt überschritten wird. Die Steuerung des Systems kann durch die Regelung des MPL-Wertes oder durch die Veränderung der Prioritätssteuerung vorgenommen werden. Die Anpassung anderer Kontrollparameter ist in dieser Stufe der Projektes noch nicht Bestandteil der Lastkontrolle.

Die Rückkopplungsschleife (feedback loop) der lokalen Lastkontrolle ist ein zyklisch zu durchlaufendes Programmmodul. Die Wahl der **Zykluszeit**, die zwischen den einzelnen Aktivierungen vergeht, muß sich an zwei Aspekten orientieren. Die Kontrolle darf zum einem keinen übermäßigen Arbeitsaufwand für das System darstellen. Auf der anderen Seite muß die

Komponente aber auch hinreichend kurzfristig auf Änderungen reagieren können. Es ist folglich ein Kompromiß zwischen **Effizienz** und **Reagibilität** zu schaffen.

Ausgangspunkt einer jeden Regelung ist die **Problemerkennung**. Diese kann durch das Feststellen einer Überschreitung der Antwortzeitziele erfolgen. Die Erkennung eines Problems kann durch die Verfolgung der zuletzt beendeten Transaktionen (TA) erfolgen. Dies macht es jedoch notwendig, Statistiken für diese Auswertungen zu führen. Um einen Kompromiß zwischen Aufwand und Nutzen zu schaffen, kann diese Informationsbeschaffung durch Stichproben erfolgen.

Bei der Problemerkennung muß bereits differenziert werden, ob alle Transaktionsklassen Leistungsprobleme aufweisen, oder ob dies nur für eine Teilmenge gilt. Ist dies für alle der Fall, so kann man davon ausgehen, daß es sich um ein generelles bzw. globales Problem handelt. Verfehlen nur eine Teilmenge der Transaktionsklassen ihre Zielvorgaben, so ist das Problem typspezifisch zu lösen. Im folgenden werden für diese beiden unterschiedlichen Kategorisierungen die Bezeichnungen **globaler und lokaler Engpaß** verwendet.

Die **Ursachenbestimmung** erfolgt durch die Analyse des Systemzustands und der Laststatistiken. Das Vorgehen bei der Analyse kann sowohl für den lokalen als auch den globalen Engpaß in ähnlicher Form erfolgen. Es bietet sich an, für diese Aufgabe bestimmte Indikatoren zu verwenden. Die Vorgehensweise bzw. Methode läßt sich an Hand der verwendeten Indikatoren in zwei Gruppen unterteilen :

(1) Erkennung des Problems aufgrund der **Antwortzeit-Zusammensetzung** der beendeten Transaktionen

Die Antwortzeiten von beendeten Transaktionen werden auf ihre Bestandteile hin geprüft. Es ist somit notwendig, für jede Transaktionsklasse die für die einzelnen Teilbearbeitungsschritte (z.B. CPU-Anteil, Sperranteil, E/A-Anteil) benötigte Zeit zu dokumentieren. Aufgrund dieser Antwortzeitbestandteile wird dann versucht, die Ursachen zu klären (z.B. durch Überschreitung eines bestimmten Schwellwertes für die einzelnen Bearbeitungsschritte).

(2) Erkennung des Problems aufgrund **systemspezifischer Indikatoren**

Für die hardware-orientierten Komponenten lassen sich Indikatoren wie z.B. die CPU- oder die Plattenauslastung bestimmen. Durch Vergleich der Auslastungsindikatoren mit extern vorzugebenen Schwellwerten können dann Schlüsse über die Art des Engpasses gezogen werden.

Für die mehr software-orientierten Komponenten müssen bestimmte Indikatoren bestimmt werden. Als Beispiel ist hier vor allem der **Sperrbearbeitung** zu nennen. Es ist notwendig zu bestimmen, welche Indikatoren verwendbar sind und auf welche Werte die Schwellwerte gesetzt werden müssen.

Als Ergebnis der Problemanalyse wird festgestellt, welche Art von Engpaß bzw. welche Menge von Engpässen vorliegt. Bei einem einzelnen Engpaß kann die Regelungskomponente ansetzen, diesen zu beheben. Bestehen jedoch mehrere Engpässe, so ist zu entscheiden, welcher Engpaß primär behoben werden kann. Bei der Lösung dieser Frage müssen die Abhängigkeiten zwischen den Engpässen berücksichtigt werden. In Abb. 3.2 werden mögliche Ursache-Wirkungsbeziehungen dargestellt.

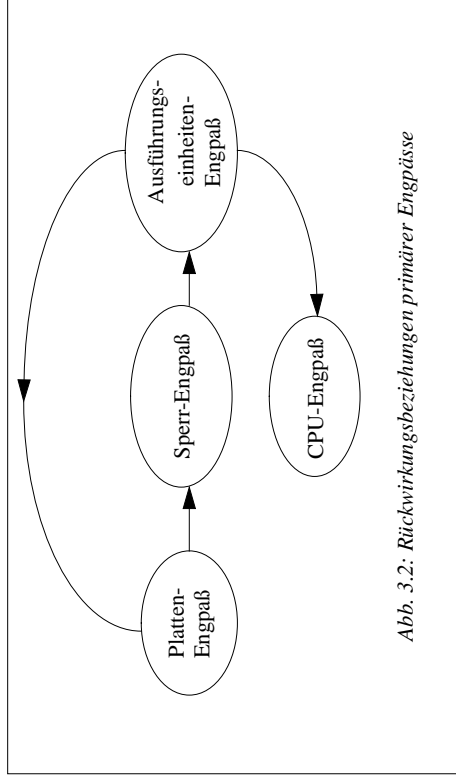


Abb. 3.2: Rückwirkungsbeziehungen primärer Engpässe

Ein Plattenengpaß kann sowohl einen Sperrengpaß, als auch einen Ausführungseinheiten-Engpaß bewirken. Denn eine Verzögerung auf der E/A-Ebene verlängert die Verweildauer der Transaktionen im System. Entsprechend verlängert sich auch die durchschnittliche Sperrhalte-

zeit der Transaktionen. Als zweite Konsequenz kann ein Plattenengpaß zu einer Verminderung der Anzahl der verarbeiteten Transaktionen pro Zeitintervall führen. Als Konsequenz ergibt sich unter der Prämisse einer gleichbleibenden (ankommenden) Last eine längere Verweildauer der Transaktionen in der Warteschlange für den Verarbeitungsbeginn. Entsprechend lassen sich auch die Beziehung zwischen Sperrengpaß und Ausführungseinheiten-Engpaß sowie CPU-Engpaß und Ausführungseinheiten-Engpaß erklären.

Lediglich der Ausführungseinheiten-Engpaß hat keine Rückwirkungen auf andere Engpaßressourcen. Durch den MPL-Wert wird festgelegt, wie viele Ausführungseinheiten gleichzeitig aktiv sein dürfen. Neu ankommende Transaktionen werden nur dann zur Verarbeitung zugelassen, wenn dieser MPL-Wert noch nicht erreicht wurde. Ist kein MPL-Slot mehr frei, so werden sie in eine Warteschlange abgestellt. Aus diesem Grunde wird im folgenden auch ein anderer Begriff für diese Art des Engpasses verwendet. Ein Ausführungseinheiten-Engpaß kann auch als **BOT-WS-Engpaß (Begin-Of-Transaction-Warteschlangen-Engpaß)** bezeichnet werden

Die Engpässe lassen sich folglich aufgrund ihrer Rückwirkungen in zwei Gruppen unterteilen. Zu der ersten Gruppe gehören alle Engpässe die Engpässe auf anderen Stufen mit sich bringen können. Die drei zu dieser Gruppe gehörenden Engpässe werden primäre Engpässe genannt. Zur Gruppe der sekundären Engpässe gehört allein der Ausführungseinheiten-Engpaß.

Den Abhängigkeiten kann bei der Erkennung von mehreren Engpässen auf drei Arten Rechnung getragen werden. Die Bestimmung des Engpasses wird an den Knoten ohne eingehende Abhängigkeits- bzw. Rückwirkungskante begonnen. Als mögliche Prüfungsreihenfolgen ergeben sich folglich:

- (1) CPU - Platten - Sperren - BOT
- (2) Platten - CPU - Sperren - BOT
- (3) Platten - Sperren - CPU - BOT

Die Engpaßprüfung beginnt links und wird nach rechts fortgesetzt, bis ein Engpaß ermittelt werden konnte. An dieser Stelle wird abgebrochen und versucht, den ermittelten Engpaß zu beheben. Für das weitere Vorgehen wurde Variante zwei gewählt. Diese Wahl wurde getroffen, da die E/A für die meisten Transaktionsklassen der Hauptbestandteil ihrer Bearbeitungszeit

darstellt. In Abb. 3.3 wird die gewählte Reihenfolge der Engpaßbestimmung noch einmal schematisch dargestellt.

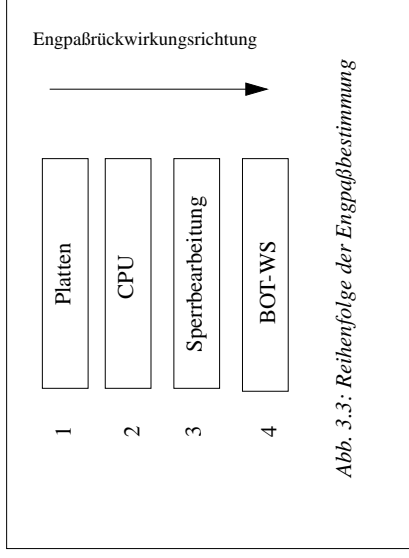


Abb. 3.3: Reihenfolge der Engpaßbestimmung

Die bisher dargestellten Überlegungen müssen bei der Erkennung eines lokalen Engpasses modifiziert werden. Für den Platten- und CPU-Engpaß existieren Einschränkungen bei der Verwendung der systemspezifischen Indikatoren. Diese Größen sind von globaler Natur. Es existieren i.d.R. keine Statistiken die den Ressourcenverbrauch der einzelnen Lastklassen für diese Bereiche dokumentieren. Es kann zwar im allgemeinen davon ausgegangen werden, daß von CPU- und Plattenengpässen alle Lastklassen betroffen sind. Die Auswirkungen können jedoch für verschiedene Transaktionstypen unterschiedlich stark ausfallen.

Bei der Analyse eines lokalen Sperrengpasses besteht die Möglichkeit, daß der Hauptverursacher für diesen Engpaß nicht die eigene Lastklasse ist. Die Regelung muß somit in der Lage sein, solche Beziehung zwischen zwei Lastklassen auch in die Steuerung mit einzubeziehen.

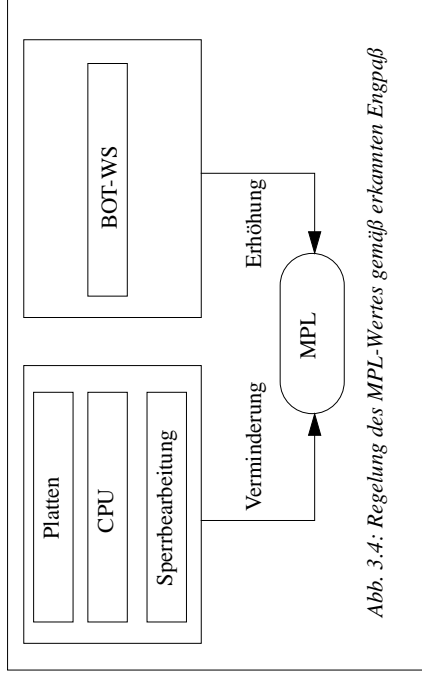


Abb. 3.4: Regelung des MPL-Wertes gemäß erkannten Engpaß

Nach der Bestimmung des Engpasses muß das System in der Lage sein, geeignete Maßnahmen zur **Engpaßbehebung** ergreifen zu können. Diese Maßnahmen können in der Veränderung des MPL-Wortes oder der Prioritäten der Lastklassen bestehen. Die drei primären Engpässe weisen auf eine zu hohe Arbeitsbelastung des Systems hin. Folglich ist es notwendig, den MPL-Wert zu senken. Lediglich im Falle der Erkennung eines BOT-WS-Engpasses ist die Arbeitslast zu niedrig und der MPL-Wert ist zu erhöhen. Es ist aber zu bedenken, daß aufgrund der gewählten hierarchischen Vorgehensweise ein BOT-WS-Engpaß nur dann erkannt werden kann, wenn **kein anderer Engpaß** erkannt worden ist. In Abb. 3.4 wird dieses Vorgehen übersichtsartig dargestellt.

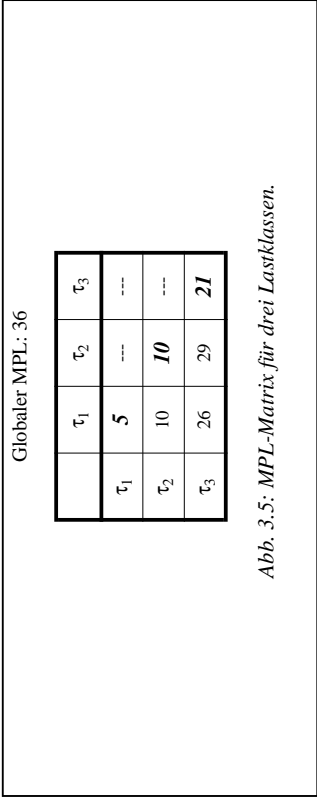
Die Veränderung der Prioritäten von Transaktionen kann sich in zwei Bereichen auswirken. Die erste Möglichkeit besteht in der Verwendung von Prioritäten bei der BOT-Behandlung. Aus der der BOT-WS werden die Transaktionen dann gemäß ihrer Prioritäten ausgewählt. Die Transaktionen mit der höchsten Priorität erhalten als erste einen freien bzw. frei werdenden MPL-Slot. Die Zuweisung erfolgt streng nach Prioritäten. Transaktionen mit niedrigeren Prioritäten werden nur dann zur BOT-Behandlung zugelassen, wenn keine andere Transaktionen mit höherer Priorität mehr in der BOT-WS warten.

Die zweite Möglichkeit einer prioritätsgesteuerten Verarbeitung ist die CPU-Warteschlange. Durch die Steuerung mit Prioritäten werden einzelnen Transaktionsklassen bevorzugt die Engpaßressourcen zugewiesen.

Es bietet sich, an die Verarbeitungserihenfolge in diesen Warteschlange durch Prioritätsalgorithmen wie FCFS (First Come First Serve), ED (Earliest-Deadline), Random (zufallsbedingte Auswahl) u.a. zu steuern.

Die Behebung **lokaler Engpässe** setzt die Möglichkeit voraus, die Laststruktur individuell steuern zu können. Zu diesem Zwecke können sowohl lokale MPL-Werte als auch Prioritäten verwendet werden. Die lokalen MPL-Werte bestimmen die Anzahl aktiver Transaktionen einer Transaktionsklasse. Die Einführung von lokalen MPL-Werten bietet folglich die Möglichkeit zu einer Last-Mix-Kontrolle.

In den Ausführungen zur Engpaßkontrolle wurden bereits die speziellen Charakteristika von lokalen Sperrengpässen erwähnt. Für lokale Sperrengpässe kann es notwendig sein, die Anzahl der aktiven Transaktionen zweier Transaktionsklassen steuern zu können. Dies kann durch Festlegung von gemischten MPL-Werten geschehen. Wird bei der Ursachenbestimmung ein lokaler Sperrengpaß für Transaktionsklasse 1, dessen Hauptversacher die Transaktionsklasse 2 ist, so besteht eine Möglichkeit in der Senkung ihres gemischten MPL-Wertes. Durch die Senkung des gemischten MPL-Wertes wird die Anzahl der parallel bearbeiteten Transaktionen der beiden konfliktträchtigen Typen im Transaktionssystem verringert.



In Abb. 3.5 wird die Verwendung von globalen, lokalen und gemischten MPL-Werten in einer Tabelle verdeutlicht. Die Werte bilden eine untere Dreiecksmatrix. Die Diagonalelemente stellen die lokalen MPL-Einstellungen dar (so ist z.B. der lokale MPL-Wert der zweiten Transaktionsklasse zehn). Der globale MPL-Wert ergibt sich als Summe der Diagonalelemente. Die gemischten MPL-Werte sind unter den Diagonalelementen eingetragen. Der gemischte MPL-

Wert der Transaktionsklasse eins und drei hat den Wert sechszundzwanzig. Dieser Wert entspricht hier dem maximal annehmbaren gemischten MPL-Wert. Er ist gleich der Summe der lokalen MPL-Werte (fünf und zehn).

Die Behebung einer BOT-WS-Engpasses birgt inhärente Probleme in sich. Die Rückwirkungen der primären Engpässe wurden bereits erörtert. Der Regelungsalgorithmus soll auch gewährleisten, daß durch die Erhöhung eines MPL-Wertes kein neuer primärer Engpaß entstehen kann. Wird diese Bedingung außer Acht gelassen, so verhält sich die Regelung wie ein **schwingendes System**, bei dem sich primäre Engpaßbehebung mit sekundären Engpaßbe-handlung abwechseln. Eine MPL-Erhöhung sollte somit nur dann gestattet werden, wenn die primären Engpaßindikatoren anzeigen, daß ein Engpaß nicht zu erwarten ist.

4. Das Simulationssystem

Im folgenden wird eine kurze Übersicht über die generellen Aufgaben der Simulationsmodule gegeben. Die anschließenden Abschnitte vermitteln dann einen Einblick in die Arbeitsweise des Simulationssystems. Die Darstellung beschränkt sich dabei auf eine rein funktionale Sichtweise. Die Darstellung der Vorgehensweisen und der verwendeten Statistiken erfolgt, insoweit dies zum Verständnis der gewonnenen Ergebnisse als notwendig erscheint. Details bezüglich der Implementation können aus der Literatur entnommen werden [Kr94, St91]. Die Implementierung des Simulationssystems wurde mit dem Simulationswerkzeug DENET (Discrete Event Network, [Li89a, Li89b]) vorgenommen. Es handelt sich dabei um ein Werkzeug zur Realisierung von diskreten, ereignisgesteuerten Simulationssystemen [EV93], das auf der Programmiersprache MODULA-2 (vgl. [Wi83]) basiert.

Im folgenden Abschnitt wird eine Übersicht über die Module des Systems gegeben. Auf diese Übersicht folgt eine Darstellung der Möglichkeiten der Lastgenerierung und des Verarbeitungsablaufs innerhalb des Systems. Im Abschnitt 4.4 wird der Regelungskreislauf beschrieben. Im letzten Abschnitt sind die durchgeführten Modifikationen des Simulationssystems dokumentiert.

4.1 Aufbau des Systems

Die Abb. 4.1 zeigt die Struktur des Simulationssystems auf lokaler Ebene bzw. für einen Verarbeitungsknoten. Der LwMgr-Modul entspricht hierbei der lokalen Lastkontrolle. Bei den anderen Modulen handelt es sich um:

- **TM :** Nach der Verteilung der Arbeit auf die einzelnen Knoten übernimmt der Transactions-Manager die Koordinierung der weiteren Verarbeitung.
- **CPU:** Dieses Modul simuliert eine Ein- oder Mehrprozessor-CPU, wobei alle Prozessen die gleiche MIPS-Leistung zur Verfügung stellen. Die Verarbeitung einer Menge von Instruktionen wird dabei als Belegen der Ressource für eine Zeitspanne simuliert, die sich aus der Anzahl der zu verarbeitenden Instruktionen und der MIPS-Leistung ergibt.
- **CC:** Die Concurrency-Control-Einheit simuliert einen Sperrverwalter, der für die Genehmigung und Freigabe von Sperren [BHG87] verantwortlich ist. Durch die Ver-

wendung von Sperren werden alle Zugriffe (Schreib- und Lesezugriffe) der TA auf die Datenbankobjekte synchronisiert. Das zugrundeliegende Sperrverfahren ist ein Zweiphasen-Sperrprotokoll (d.h. eine Transaktion hält die ihr gewährten Sperren bis zum Transaktionsende) mit zwei Sperrmodi (Read- und Exclusive-Sperren). Lesesperren können an mehrere Transaktionen parallel gewährt werden. Schreibsperren werden immer nur an eine Transaktion vergeben. Als Sperrgranulat können partitionsbezogenen Seiten oder Objekte gewählt werden. Die Verwendung von Sperren dient der Zugriffssynchronisation. Auf die Synchronisation kann in den Simulationen auch verzichtet werden.

- **DL:**

Eng an den CC-Modul gekoppelt ist der Deadlock-Modul. Für die Transaktionen die Sperren halten, und die Transaktionen, die auf bereits vergebene Sperren warten, wird ein Wartegraph geführt (vgl. [Ta85]). In diesem Wartegraph werden die Beziehungen zwischen Sperrinhabern und Sperranforderern als gerichtete Kanten eingetragen. Der Modul übernimmt die Deadlock-Erkennung (unter einem Deadlock versteht man eine „Systemverklemmung“ aufgrund zyklischer Wartebeziehungen zwischen Transaktionen), indem sie den Wartegraphen auf Zyklen untersucht und ggf. den Kreis durch Abbruch einer Transaktion aus dem Kreis unterbricht.

- **BM:**

Der Buffer-Manager realisiert die Pufferverwaltung und stellt die Schnittstelle zur Peripherie dar. Die Ausschreibstrategie kann durch Simulationsparameter verändert werden; möglich ist die Simulation einer FORCE oder NOFORCE Strategie [GR93]. Als Seitenverdrängungsalgorithmus wird ein einfacher LRU-Algorithmus (Least Recently Used - Algorithmus) verwendet [SPG91].

Das Logging der Pufferverwaltung erfolgt ohne Zwischenspeicherung unmittelbar auf die Platte; das Ausschreiben der Logging-Daten erfolgt synchron. Pro Änderungstransaktion wird ein Schreibvorgang auf die Log-Platte notwendig. Die Simulationen können auch ohne Logging durchgeführt werden.

Die Peripherie wird durch folgende Module simuliert:

- **IO:** Dieses Modul delegiert die vom BM-Modul angeforderten Plattenzugriffe an die Externspeicher.
- **LOGDISK,**
- DISK:** Eine Instanz kann jeweils mehrere Platten modellieren, wobei nach DB- und Log-Platten unterschieden wird. Die Annahme von Aufträgen wird über Controller abgewickelt, die die Zugriffe auf die Platte weiterleiten.

Neben denen in Abb. 4.1 gezeigten Hauptmodulen existieren noch Hilfsmodule, die der Beschreibung der Systemkonfiguration, der Realisierung einer eigenen Freispeicherverwaltung oder der Definition von globalen Datentypen dienen.

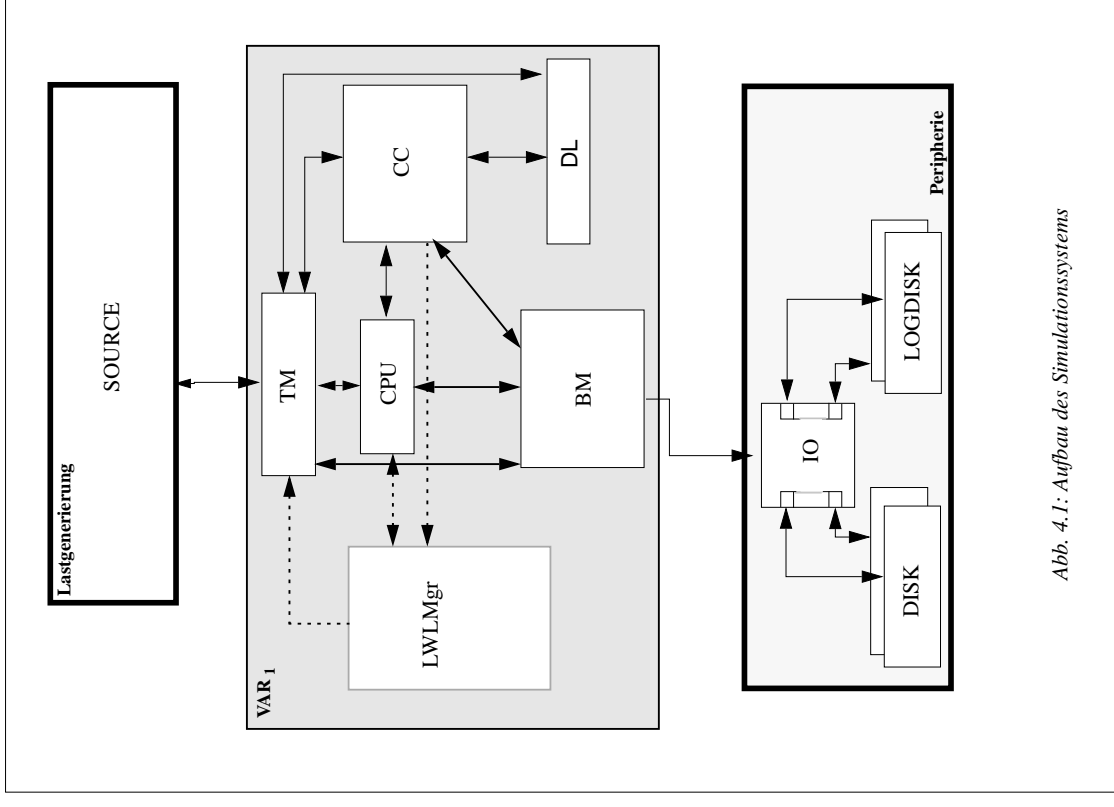


Abb. 4.1: Aufbau des Simulationssystems

4.2 Modellierung der Last

Das vorliegende System unterstützt eine vierstufige Hierarchie von Datenobjekten :

- (1) Datenbank
- (2) Partition
- (3) Seite
- (4) Objekt

Die Datenbank besteht aus M Partitionen. Jede dieser Partitionen enthält eine bestimmte Anzahl von Seiten. Durch Angabe eines Blockungsfaktors pro Partition wird die Anzahl der Objekte pro Seite festgelegt. Aus diesen Angaben läßt sich dann die gesamte Anzahl der Objekte pro Partition bestimmen.

Das System unterstützt zwei Arten der Lasterzeugung: Die erste Art der Erzeugung ist die Generierung einer **synthetischen Last**. Die andere Möglichkeit basiert auf dem Mitschnitt einer **realen Datenbanklast**, die in das bestehende System eingelesen werden kann. Die Möglichkeiten der Variation der realen Last sind hierbei jedoch beschränkt. Es besteht lediglich die Möglichkeit, die Ankunftsrate zu variieren. Hierdurch läßt sich dann eine Raffung oder Streckung der Last bzgl. der Zeitachse bewirken. Die Möglichkeit der Verwendung einer realen Datenbanklast erlaubt andererseits die Bewertung der Regelungsstrategien realitätsnah durchzuführen. Die Raffung und Streckung der Last kann dazu verwendet werden, die auftretenden Engpässen über einen kürzeren oder längeren Zeitraum zu verteilen. Die Engpaß-Situationen werden somit verstärkt oder abgeschwächt.

Für eine synthetische Last sind eine Reihe von Eingabeparametern festzulegen. Pro Transaktionstyp muß festgelegt werden:

- (1) mittlere Ankunftsrate [Transaktion pro Sekunde (tps)]
- (2) mittlere Anzahl der Referenzen
- (3) mittlerer Prozentsatz der Schreibreferenzen bzgl. aller Referenzen
- (4) die Verteilung der Zugriffe auf die verschiedenen Partitionen

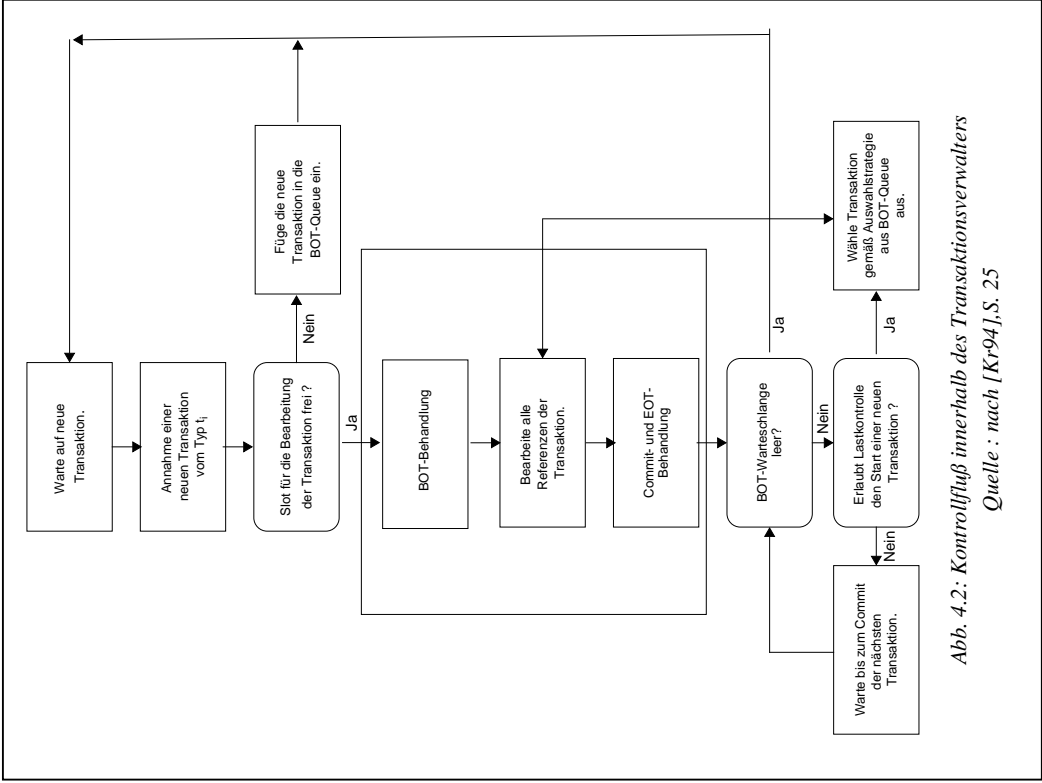
Das Ankunftsverhalten der Transaktionen muß nicht „starr“ um die mittlere Ankunftsrate schwanken. Die Ankunftsreihe kann durch Kombination einer Grundlast und der Angabe von Zeitintervallen für eine höhere Last flexibel gestaltet werden. Im folgenden werden die Lasten mit nur einer Ankunftsrate als **stabile Lasten** und die Lasten mit Lastspitzen als **variable Lasten** bezeichnet. Das System bietet auch die Möglichkeit, die Anzahl der Referenzen entweder als konstanten Wert oder als Erwartungswert einer Exponentialverteilung aufzufassen. Von der letzteren Möglichkeit wird aber in den Simulationen kein Gebrauch gemacht. Für die Transaktionsklassen kann festgelegt werden, auf welche Partitionen sie zugreifen. Dies geschieht in Form prozentualer Angaben. Diese Prozentzahlen ergeben zusammen mit der Anzahl der Referenzen einer Klasse die Quantität der Referenzen für eine Partition und eine Transaktionsklasse. Geht man z.B. davon aus, daß die Datenbank aus zwei Partitionen besteht, so gibt eine Zugriffsverteilung der Form (0.1,0.9) für die Transaktionsklasse i an, daß 10 % der Referenzen Partition eins und 90 % Partition betreffen.

4.3 Der Transaktionsverwalter

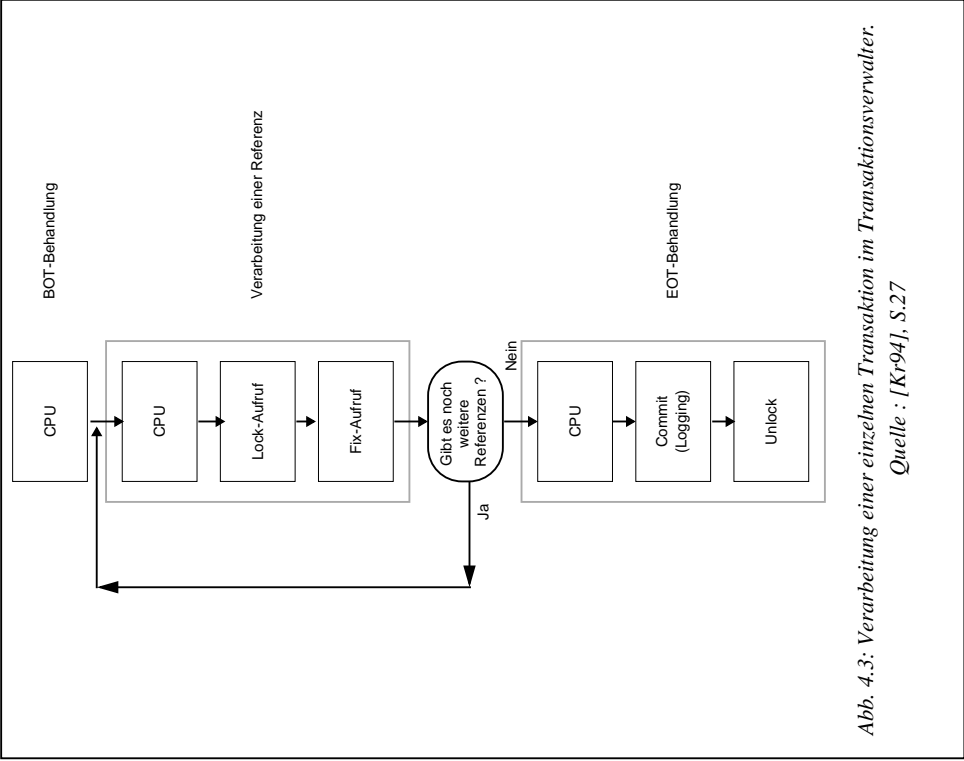
Der Transaktions-Manager ist eine zentrale Komponente innerhalb des Simulationssystems. Seine Aufgaben umfassen die Annahme der Transaktionen und die Kontrolle bzw. Koordination der einzelnen Verarbeitungsschritte. Abb. 4.2 gibt einen Überblick über den Verarbeitungsablauf. Der Transaktions-Manager koordiniert die Aufrufe der einzelnen Komponenten des Simulationssystems. Zusammen mit dem Sperr-Manager, der CPU und dem Plattenmodul sammelt er wesentliche Daten zum augenblicklichen Systemzustand. Diese Daten liefern einen wichtigen Beitrag für die Entscheidungen und Handlungen der Lastkontrolle.

Der Start der Verarbeitung wird durch die Festlegung des MPL-Wertes und durch die augenblickliche Lastsituation geregelt. Eine neu ankommende Transaktion wird nur dann direkt zur Verarbeitung freigegeben, wenn noch eine freier MPL-Slot vorhanden ist. Ist dies nicht der Fall, so wird die Transaktion in die Warteschlange des TM eingereiht.

In Abb. 4.3 wird die Verarbeitung einer einzelnen Transaktion dargestellt. Nach der BOT-Behandlung wird für jede Datenreferenz ein CPU-Vorlauf vorgenommen. Falls mit Synchronisation gearbeitet wird, folgt diesem dann die Sperr-Anforderungen. Nach Gewährung der Sperre wird das angeforderte Objekt durch einen FIX-Aufruf an die Pufferverwaltung in



Hauptspeicher des Verarbeitungsrechners bereitgestellt. Nach der Bearbeitung erfolgt dann ggf. noch das Logging und danach werden die angeforderten Sperren wieder freigegeben.



4.4 Der Regelungskreislauf

Die Lastkontrolle kann in zwei verschiedenen Betriebsmodi operieren. Im Betriebsmodus mit globaler MPL-Regelung (GMPL-Regelung) wird nur eine Anpassung bzw. Variation des globalen MPL-Wertes vorgenommen. Im anderen Modus wird zusätzlich ein lokaler MPL-Wert verwendet (LMPL-Regelung). In Abhängigkeit von der erkannten Engpaß-Situation (lokal/global) werden bei der LMPL-Regelung die lokalen MPL-Werte oder nur der globale MPL-Wert verändert. Bei beiden Betriebsmodi besteht die zusätzliche Möglichkeit der Verwendung von Prioritäten.

Die Lastkontrolle wird in periodischen Abständen aufgerufen, unabhängig davon, ob die Lastsituation im Transaktionssystem als kritisch oder normal zu bewerten ist. Diese periodischen Abstände können beim Vorliegen bzw. Erkennen einer Problemsituation auch verkürzt werden. Die Möglichkeit der Verkürzung des Intervalls kann dazu beitragen die Reagibilität der Regelung zu erhöhen. Von dieser Option wurde in den Simulationen jedoch keine Verwendung gemacht. Nach Aktivierung führt die Lastkontrolle eine Bewertung der Lastsituation durch. Das alleinige Kriterium für diese Bewertung ist die Einhaltung der transaktionstypspezifisch vorgegebenen Antwortzeitziele. Werden diese Antwortzeiten eingehalten, so unternimmt die Lastkontrolle nichts. Sie unternimmt auch dann nichts, wenn die Auslastungsindikatoren (wie CNFL-Rate, CPU-Auslastung, etc.) auf Engpässe hinweisen, die Antwortzeitziele jedoch eingehalten werden.

Nach der Erkennung einer Überlastsituation, erfolgt der zweite Schritt bei der Regelung des Transaktionssystems: Das Lastkontrollmodul unterscheidet zwischen einem globalen und einem lokalen Engpaß. Ein **globaler Engpaß** liegt vor, wenn ein wesentlicher Anteil der Transaktionsklassen seine individuellen Zielvorgaben verfehlt hat. Ein **lokaler Engpaß** liegt vor, wenn nur wenige Typen ihre Zielantwortzeit nicht einhalten konnten. Der Anteil an Typen, die ihre Zielvorgaben verfehlen müssen, damit eine globaler Engpaß erkannt wird, kann individuell für jeden Simulationslauf festgelegt werden. Der Wert ist jedoch eine Konstante während des Simulationslaufs. In der nächsten Phase unternimmt das Regulationssystem eine Ursachenanalyse.

Im dritten Schritt verändert das Lastregelungssystem die Steuerungsparameter für das Transaktionssystem. Dies geschieht unter Berücksichtigung der erkannten Krisensituation (lokal, global) und der ermittelten Ursache für die Krisensituation. Bei lokalen Engpässen wird eine

typespezifische Veränderung der MPL-Werte vorgenommen oder die Priorität eines Transaktionstyps verändert.

Bei globalen Engpässen modifiziert das Lastregelungssystem in erster Linie den globalen MPL-Wert. Je nach gewählter Regelungsstrategie werden entweder alle lokalen Werte verändert oder ausschließlich der globale MPL-Wert.

Für die Aktivierung der Lastkontrollkomponente wird im Simulationsmodell CPU-Zeit berechnet. Die berechnete Zeit ist abhängig davon, ob ein Engpaß erkannt wurde oder eine unkritische Lastsituation vorliegt.

4.4.1 Bewertung der Lastsituation

In den vorigen Kapiteln wurde vereinfacht davon ausgegangen, daß die Bewertung des Lastsituation durch den Vergleich der Antwortzeiten einzelner Transaktionsklassen mit den vorgegebenen Zielen vorgenommen wird. In [Kr94] wurden drei unterschiedliche Verfahren zur Bewertung der Lastsituation implementiert, die eindeutige Informationen darüber liefern sollen, ob das Antwortzeit-Verhalten „gut“ oder „schlecht“ ist.

Bei allen Verfahren wird die Bewertung zweistufig durchgeführt: Im ersten Schritt werden die einzelnen Transaktionsklassen individuell abgeprüft. Für jeden Typ wird dann entschieden, ob die Zielantwortzeiten eingehalten worden sind. Ist dies nicht der Fall, so wird die entsprechende Lastklasse als kritisch bewertet. Ist im Simulationssystem nur die GMPL-Regelung eingeschaltet, so genügt **eine kritische Lastklasse** für den Start der Engpaßbehandlungsregeln. Im Fall mit lokaler MPL-Regelung kann vorgegeben werden, wie viele Transaktionsklassen als kritisch bewertet sein müssen, bis von einem globalen Problem ausgegangen werden soll. Ist dieser Wert noch nicht erreicht, so wird von einem lokalen Problem ausgegangen.

Implementiert wurden im Simulationssystem die folgenden Metriken :

(1) Vorgabe von individuellen Anteilen für die überfälligen Transaktionen

Es werden sowohl alle laufenden als auch die Transaktionen, die in jüngster Vergangenheit (im Zeitraum zwischen der Regelungsläufen) beendet wurden, auf die Verfehlung ihrer Antwortzeit untersucht. Zielvorgaben sind für die Transaktionsklassen vor-

gegebene Antwortzeiten. Für jede der n Lastklassen wird der Anteil f'_i der Transaktionen bestimmt, die ihr Ziel verfehlt haben. Dieser Anteil wird mit einem extern vorgegebenen Anteil f_i verglichen. Ist der tatsächliche Anteil f'_i größer, dann wird die entsprechende Lastklasse als „kritisch“ gewertet.

(2) Prüfung der Subgoals der laufenden Transaktionen

Die laufenden Transaktionen werden auf die Verfehlung eines Subgoals untersucht. Dieses Subgoal berechnet sich aus dem Quotienten des Anteils der bereits verbrauchten Bearbeitungszeit von den vorgegebenen Ziel-Antwortzeiten und dem Anteil der bereits bearbeiteten Seitenreferenzen. Um diesen Anteil berechnen zu können, wird für jeden Transaktionstyp die Anzahl der zu bearbeitenden Referenzen geschätzt. Diese Schätzung wird durch die Anzahl der bearbeiteten Seiten von bereits beendeten Transaktionen dieses Typs vorgenommen. Diesem Verfahren liegen somit zwei Annahmen zugrunde. Zum einem wird von einer relativ homogenen Referenzzahl innerhalb einer Transaktionsklasse ausgegangen. Zum anderen wird angenommen, daß die Bearbeitungszeit linear zur Anzahl der Referenzen ist.

(3) Prüfung der mittleren Antwortzeiten der abgeschlossenen Transaktionen

Es werden die mittleren Antwortzeiten der Transaktionen einer Lastklasse mit dem vorgegebenen Zielvorgaben für die Klasse verglichen. Eine Transaktionsklasse wird als kritisch gewertet, wenn die mittlere Antwortzeit größer als die Zielvorgabe ist.

Alle diese Verfahren sind mit Vor- und Nachteilen behaftet. So ist eine Verwendung von laufenden Transaktionen immer mit spekulativen Annahmen behaftet. Eine Diskussion dieser Problematik wurde in [Kr94] vorgenommen.

In den in dieser Arbeit durchgeführten Simulationen wurde das dritte Verfahren gewählt. Die Lastkontrollkomponente versucht somit, die **mittleren Antwortzeiten der Transaktionsklassen** unter den Zielvorgaben zu halten.

4.4.2 Ursachenbestimmung

Wurde im ersten Schritt des Regelungsverfahrens eine lokale oder globale Überlastsituation erkannt, erfolgt im zweiten Schritt die Ursachenbestimmung. Im ursprünglichen System [Kr94] werden die Antwortzeitanteile der einzelnen Verarbeitungsschritte (Platten, Sperrwartzeit, CPU-Bearbeitungszeit, Wartezeit in der Eingangswarteschlange) analysiert. Nur in dem Fall, in dem keine Antwortzeiten zur Verfügung stehen, werden die Auslastungsindikatoren zur Ursachenbestimmung herangezogen. Diese Variante der Ursachenbestimmung wird im folgenden antwortzeitbezogene Engpaßerkennung genannt.

Daneben wird hier auch eine rein auslastungsbezogene Ursachenbestimmung unterstützt. Bei dieser Erkennungsvariante werden nur die Auslastungsindikatoren zur Ursachenbestimmung verwendet. Eine Analyse der Antwortzeit wird nicht vorgenommen.

Bei lokalen Engpässen wird die Zusammensetzung der Antwortzeiten derjenigen Transaktionsklasse τ_i untersucht, die ihre Zielvorgaben nicht einhalten haben. Je nach Gewicht der einzelnen Anteile der Antwortzeit (s. Abschnitt 4.4.2.1) wird dann im zweiten Schritt versucht, eine individuell auf den jeweiligen Typ abgestimmte Verbesserung der Situation zu erreichen. Bei der Untersuchung der Antwortzeit und/oder der Auslastungsindikatoren wird zwischen globalen und lokalen Engpässen unterschiedlich vorgegangen. Bei globalen Engpässen werden bei antwortzeitbezogener Engpaßerkennung die gemeinsam erfaßten Antwortzeiten aller beendeten Transaktionen verwendet. Entsprechend werden bei der anderen Variante die globalen Auslastungsindikatoren verwendet. Bei einem lokalen Engpaß werden nur die Antwortzeiten der problembehafteten Transaktionsklasse untersucht. Ein solches Verfahren ist bei den Auslastungsindikatoren nur eingeschränkt möglich, da ein Auslastungsindikator für ein einzelne Transaktionsklassen nur für Sperrkonflikte implementiert ist. Weitere Einzelheiten zu der verwendeten Indikatoren und Berechnungen finden sich im Abschnitt 4.4.2.1.

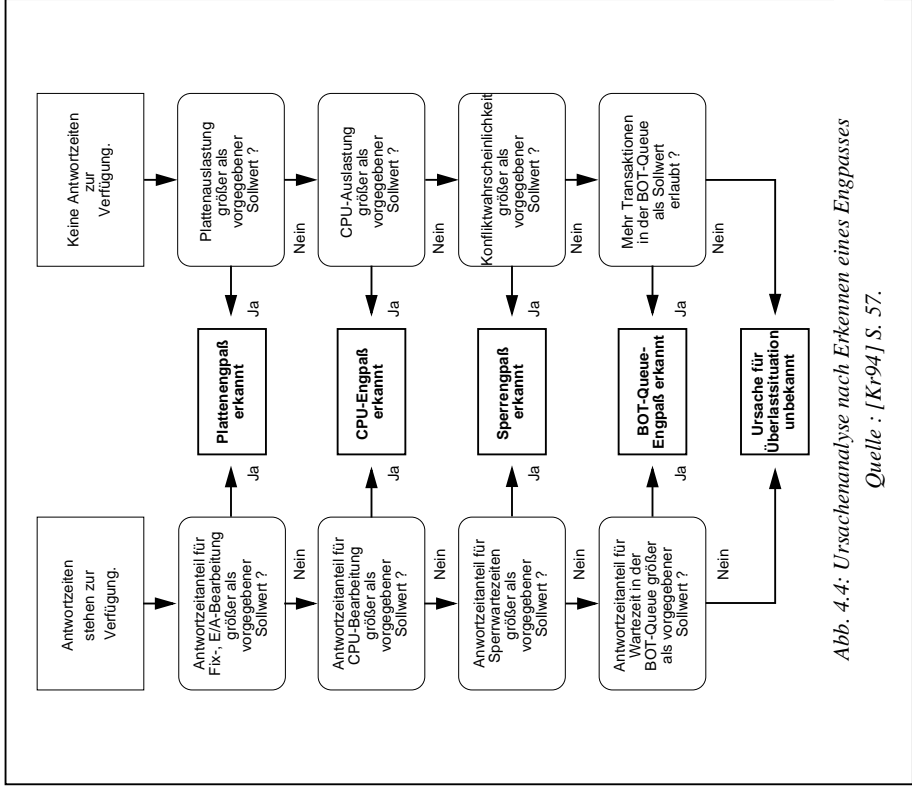


Abb. 4.4: Ursachenanalyse nach Erkennen eines Engpässes
Quelle : [Kr94] S. 57.

Abb. 4.4 stellt die Analysehierarchie des Lastkontrollmoduls dar. Die primären Engpässe wie Platten-, CPU-, und Sperrengpaß werden zuerst abgeprüft. Der Kontrollfluß der Erkennungsroutine wird unterbrochen, sobald ein Engpaß erkannt wird. Die Abb. 4.4 zeigt die Variante der antwortzeitbezogenen Erkennung. Die Erkennung mittels Auslastungsindikatoren kann man durch das Ausblenden des linken Teilstrangs erhalten.

4.4.2.1 Die Parameter für die Ursachenerkennung

Bereits im Abschnitt 2 wurden die grundlegenden Möglichkeiten der Parameterwahl erläutert. In diesem Abschnitt sollen weitere Erläuterungen zu den beiden Gruppen gemacht werden. Diese Informationen sind notwendig, da die Ergebnisse aus den Kapiteln 5 und 6 möglicherweise zu irrtümlichen Schlüssen führen könnten.

Die Parameter der Antwortzeitanalyse.

Im System werden pro Transaktionsklasse Statistiken geführt, die Aufschluß über die absoluten Antwortzeiteile für die einzelnen Bearbeitungsschritte ermöglichen. In Abb. 4.5 sind die einzelnen Anteile der Antwortzeit dargestellt.

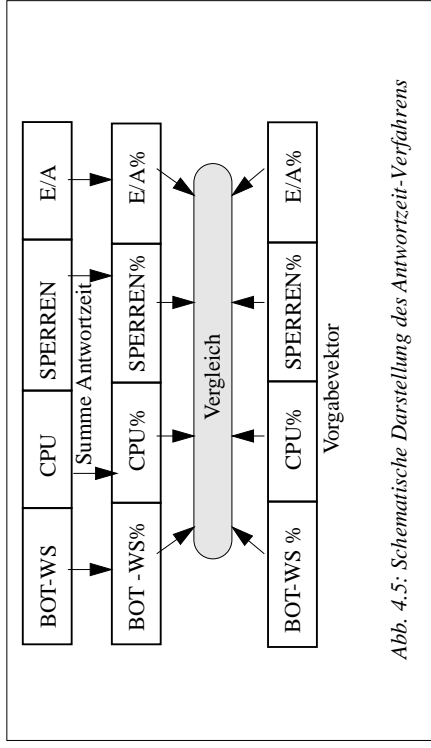


Abb. 4.5: Schematische Darstellung des Antwortzeit-Verfahrens

Zur Analyse der Antwortzeiten werden zu jedem Regelungszeitpunkt die Mittelwerte des letzten Verarbeitungszklus verwendet. Das Simulationssystem bietet auch Alternativen zu diesem Vorgehen (Verwendung anderer Metriken). In den Simulationen werden jedoch nur diese **vergangenheitsorientierte Werte** verwendet. Um die Vergleiche über Transaktionsklassen hinweg zu ermöglichen, werden bei der Analyse die absoluten Antwortzeiteile in Bezug zur gesamten Antwortzeit gesetzt. Bei der globalen Engpaßerkennung werden die für jede einzelne Transaktionsklasse ermittelten Prozentsätze addiert und durch die Anzahl der Klassen dividiert.¹ Die Analyse erfolgt durch den Vergleich der Prozentsätze mit vorgegebenen Grenzwerten.

werten. So wird z.B. ein Plattenengpaß dadurch erkannt, wenn der prozentuale Antwortzeitanteil der in der letzten Periode beendeten Transaktionen größer als 90% ist.

In Tabelle 4.1 sind die Parameter aufgeführt die in der zugrundeliegenden Diplomarbeit [Kr94] für die Simulationen benutzt wurden.

Schwellwerte für zeitliche Anteile:	Wert	Schwellwerte für Auslastungsindikatoren:	Wert
E/A-Wartezeitanteil:	90 %	Schwellwert für das Erkennen einer E/A-Überlast (Plattenauslastung):	50 %
CPU-Bearbeitungszeitanteil:	25 %	Schwellwert für das Erkennen einer CPU-Überlast (CPU-Auslastung):	50 %
Sperr-Wartezeitanteil:	25 %	Schwellwert für das Erkennen eines Sperr-Engpasses: 1. Konfliktwahrscheinlichkeit 2. WKM-Rate	1 % 1.3
BOT-Queue-Wartezeitanteil:	10 %	Schwellwerte für das Erkennen eines BOT-Queue-Engpasses (TM-Auslastung): (Anzahl der Transaktionen in der BOT-Queue):	60 % 40
Summe der zeitl. Anteile:	169 %		

Tabelle 4.1: Schwellwerte für die Lastregelungskomponente.

Diese Werte wurden als Startwerte für die Simulationen übernommen, um eine gewisse Kontinuität zu bereits in [Kr94] vorgenommenen Simulationen zu wahren. Im Rahmen der Simulationen wurden sie jedoch in vielen Fällen verändert oder variiert. Die Summe aller Anteile ist größer als 100%. Durch diese Art des *Overfitting* soll gewährleistet werden, daß Engpässe erst bei wesentlichen Abweichungen von der *Norm (100%)* erkannt werden.

Der Wert der TM-Auslastung wird bei der eigentlichen Modifikation des MPL-Wertes verwendet. Seine Verwendung wird im folgenden Abschnitt erläutert.

1. Eine Modifikation dieses Algorithmus wird in der nächsten Version des Simulationssystems vorgenommen.

Die Parameter der Auslastungsindikatoren.

In der Tabelle 4.1 sind auch die Parameter für die auslastungsbezogene Engpaßerkennung dargestellt. Die Werte für die CPU- und die Platten-Auslastung können aus den einzelnen Modulen abgefragt werden. Für den BOT-WS-Engpaß wird vereinfacht angenommen, daß er ab einer bestimmten Länge der BOT-WS vorliegt.

Die Erkennung eines Sperrengpasses muß durch neu eingeführte Maße realisiert werden. In der Literatur werden verschiedene Metriken bzw. Verfahren zur Sperrengpass-Behandlung genannt [MW92, CKL89, Th92]. Im Simulationssystem wurden zwei verschiedene Maße implementiert. Dabei handelt es sich um die **Konfliktwahrscheinlichkeit** (CNFL-Rate) und die **Konfliktrate** (WKM-Rate) [MW92]. Die Konfliktwahrscheinlichkeit wird für jede Transaktionsklasse einzeln geführt, während die WKM-Rate nur für das gesamte System berechnet werden kann. Erstere setzt die Anzahl der Konflikte zur Anzahl aller Referenzen des letzteren Regelungsintervalls in Beziehung:

$$Konfliktwahrscheinlichkeit = \frac{Anzahl\ Konflikte}{Anzahl\ Sperranforderungen}$$

Bei der WKM-Rate gehen die Anzahl der gehaltenen Sperren von aktiven und blockierten Transaktionen in die Berechnung ein. Zu den aktiven Transaktionen werden alle Transaktionen gezählt, die momentan nicht auf eine Sperre warten müssen. Es werden die Anzahl aller gewährten Sperren im System zur Anzahl der von aktiven Transaktionen gehaltenen Sperrereferenzen ins Verhältnis gesetzt.

$$Rate = \frac{Anzahl\ aller\ Sperren\ im\ Datenbanksystem}{Anzahl\ der\ Sperren\ aktiver\ Transaktionen}$$

4.4.3 Regelung des Transaktionssystems

Nach einer erkannten Überlastsituation und der nachfolgenden Ursachenanalyse wird das Transaktionssystem im dritten Schritt des Regelungsverfahrens neu eingestellt. Dazu erhält die Regelungskomponente eine Information über die **Art der Überlastsituation** (lokal/global) und eine Information über die im zweiten Schritt ermittelte **Hauptursache** für die Überlastsituation. Sind für mehrere Transaktionstypen lokale Engpässe aufgetreten, so werden diese Engpässe **nacheinander getrennt** behandelt. Für jede der Lastklassen wird in diesem Fall eine eigene Ursache ermittelt. Gegebenenfalls unterbleibt trotz eines erkannten Engpasses nach der Prüfung weiterer Informationen eine neue Einstellung des Transaktionssystems. Ist zu erwarten, daß eine Neueinstellung der Steuerungsgrößen zu anderen Engpässen führt, so wird nichts unternommen. Diese Vorgehen soll ein „schwingendes“ Verhalten der Regelungskomponente vermeiden.

Die Steuerungskomponente hat grundsätzlich zwei Eingriffsmöglichkeiten: die Veränderung der Einträge der MPL-Matrix und die Veränderung der typspezifischen Prioritäten. Ein Abbruch von laufenden Transaktionen zur Verringerung der Last wird nicht vorgenommen.

Veränderung der MPL-Werte.

Die Veränderung der MPL-Werte ist parametrisiert (relative Veränderung zur Senkung und zur Vergrößerung des MPL-Wertes) und kann im Simulationssystem eingestellt werden. Durch verschiedene Kombinationen der beiden relativen Parameter können unterschiedliche Regelungsstrategien eingestellt werden.

Relative Veränderung:	Wert:	Absolute (minimale) Veränderung:	Wert:
Senkung eines MPL-Wertes:	5 %	Senkung eines MPL-Wertes:	1
Erhöhung eines MPL-Wertes:	5 %	Erhöhung eines MPL-Wertes:	1

Tabelle 4.2: Werte zur Veränderung des MPL-Wertes

Die Angabe für die Erhöhung und die Verminderung der MPL-Werte wird in Prozentangaben angegeben. Durch die Wahl dieser Parameter wird eingestellt, wie behutsam oder stark reagiert werden soll. Die MPL-Veränderungs-Parameter werden durch Angabe eines minimalen absoluten Veränderungsgröße vervollständigt. Diese Parameter werden verwendet, wenn die auf prozentualer Basis und dem augenblicklichen MPL-Wert beruhende prozentuale Verände-

rung kleiner ist als die minimalen absoluten Veränderungsgrößen. Die in den Simulationen verwendeten Werte sind in Tabelle 4.2 wiedergegeben.

Die MPL-Werte werden folgendermaßen verändert:

- Wurde ein **Plattenengpaß** erkannt, so werden die Einträge für den globalen MPL-Wert gesenkt. Wurde als Regelungsstrategie LMPL-Regelung eingestellt, erfolgt zusätzlich eine Veränderung der lokalen MPL-Werte.
- Bei einem globalen **CPU-Engpaß** werden die Einträge für die MPL-Werte ebenfalls gesenkt. Bei LMPL-Regelung werden zunächst alle lokalen MPL-Werte verringert; anschließend wird ein neuer globaler MPL-Wert aus den lokalen Werten berechnet.
Auch bei einem globalen **Sperrengpaß** werden die Einträge der MPL-Matrix für alle Typen (lokal und globale Regelung) bzw. der globale MPL-Eintrag (nur globale Regelung) gesenkt. Bei einem lokalen Sperrengpaß wird dagegen zunächst die Konfliktmatrix (die Einträge beinhalten Kofizikraten) des Sperrverwalters untersucht: Für jeden problematischen Transaktionstyp τ_i wird derjenige Typ τ_j ermittelt, der für die hohen Sperrwartezeiten von τ_i verantwortlich ist. Gilt $\tau_i = \tau_j$, wird der lokale MPL-Wert für τ_i verringert (τ_i blockiert sich selbst). Gilt hingegen $\tau_i \neq \tau_j$, wird der gemischte MPL-Wert für τ_i und τ_j gesenkt, um die gegenseitigen Behinderung zu beschränken.

- Bei einem **BOT-WS-Engpaß** muß die Lastsituation vor der MPL-Erhöhung untersucht werden, ob andere Bereiche nicht schon nahe an der Sättigungsgrenze betrieben werden. Analog zur eigentlicher Ursachenerkennung gibt es die prinzipielle Möglichkeit der antwortzeitbezogenen und der auslastungsbezogenen Erkennung.

Beim antwortzeitbezogenen Verfahren wird zunächst der E/A-Bearbeitungszeitanteil, die Plattenauslastung, der CPU-Bearbeitungszeitanteil, die CPU-Auslastung und der Sperrwartezeitanteil geprüft. Nur wenn zwischen diesen Werten und den kritischen Schwellwerten der Abstand ausreichend groß ist, wird der MPL-Wert erhöht. Die Einstellungen für diese Schwellwerte können durch verschiedene Konstanten variiert werden. Die Definition dieser Abstände wird statisch zu Beginn der Simulation in Prozentform vorgenommen. So ergeben sich z.B. bei einer maximal erlaubten CPU-Auslastung von 100% und einem Sicherheitsintervall (im folgenden gelegentlich auch Korridor genannt) von 5% ein bei einer MPL-Erhöhung maximal erlaubter Wert von 95%.

Die ausschließliche Verwendung des Sperrwartezeitanteils stellte sich bei den Simulationen als unzureichend heraus. Das Simulationssystem wurde deshalb so modifiziert, daß sowohl die Antwortzeit als auch die Konflikttrate geprüft werden.

Beim auslastungsbezogenen Verfahren werden generell keine Antwortzeitanteile beim Abprüfen verwendet.

Bei einer Senkung eines lokalen oder des globalen MPL-Wertes wird zunächst die „Auslastung“ dieses MPL-Wertes geprüft. Die Auslastung wird aus dem Quotienten der Anzahl der in Bearbeitung befindlichen (nach der BOT-Behandlung) Transaktionen und dem momentanen MPL-Wert berechnet. Wenn die so berechnete Auslastung gering ist (z.B. unter 60%), so wird der MPL-Wert nicht um einen bestimmten Prozentsatz gesenkt, sondern um $((1 - \text{Auslastung})/2) \%$ (Cut-Regel). Geht man z.B. davon aus, daß der momentane MPL-Wert bei 100 liegt und eine Engpaß erkannt wird, der eine MPL-Reduktion bewirkt. Falls zum Regelungszeitpunkt lediglich 40 Transaktionen im System (40% MPL-Wert-Auslastung) sind, so wird der MPL-Wert um 30% auf 70 gesenkt.

Dieses Vorgehen gewährleistet einen schnelle Anpassung der MPL-Werte bei zu hoher Anfangseinstellung. In den Simulationsläufen wird zu Beginn ein MPL-Startwert extern vorgegeben.

Verwendung von Prioritäten.

Prioritäten können in zwei Bereichen eingesetzt werden. Bei diesen Bereichen handelt es sich zum einem um die BOT-WS und zum anderen um die CPU- bzw. CPU-Request-Warteschlange. Die Prioritätssteuerungen sind insoweit von einander unabhängig, daß für die einzelnen Warteschlangen auch unterschiedliche Steuerungen verwandt werden können. Die Prioritäten der Transaktionen bzw. der Transaktionsklassen sind jedoch einheitlich für beide Bereiche. Es existieren im Simulationssystem verschiedene Möglichkeiten der Prioritätsangabe.

Als Prioritätsalgorithmen können im System verwendet werden :

- First-Come-First-Serve (FCFS oder FIFO)
Die Transaktionen (bzw. ihre CPU-Requests) werden in Ankunftsreihenfolge aus den Warteschlangen ausgereiht und gestartet.
- Earliest-Deadline
Die Earliest-Deadline-Methode bestimmt für jede einzelne Transaktion ihre Deadline. Diese wird durch den Start- bzw. Ankunftszeitpunkt und die Zielantwortzeiten der jeweiligen Transaktionsklassen berechnet. Die Priorität der Transaktion wächst, je näher der vorgegebene Beendigungszeitpunkt rückt.
- Random
Die Auswahl der Transaktionen erfolgt durch zufällig.
- manuelle Vorgabe je Transaktionsklasse
- Performance-Index

Die Regelung mit Hilfe des Performance-Indexes stellt einer Erweiterung des Systems dar.

Es handelt sich beim Performance-Index um eine zielorientierte dynamische gruppenspezifische Prioritätssteuerung. Je nach dem Verhältnis zwischen den Ziel-Antwortzeit und den zuletzt erzielten Ergebnissen wird eine neue Priorität berechnet, die sowohl für die Transaktionen, die noch im System sind, als auch für die neu ankommenden Transaktionen gilt. Die Berechnung des momentanen Performance-Index (PI) einer Transaktionsklasse τ wird nach [FNGK93] berechnet durch :

$$PI_{\tau} = \frac{\text{mittlere Antwortzeit}_{\tau}}{\text{Zielantwortzeit}_{\tau}}$$

Die berechneten PI für jede Transaktionsklasse werden auf die 60 vorgesehenen Prioritätsstufen abgebildet. Die Neuberechnungen der PI finden immer zu Beginn der Aktivierung der Lastkontrolle statt.

4.5 Modifikationen des Simulationssystems

Im Rahmen der Arbeit mit dem Simulationssystem wurden Erweiterungen und Modifikationen des Systems vorgenommen. Es wurden folgende Arbeiten durchgeführt:

- Einführung der auslastungsbezogenen Erkennung
- Einführung der WKM-Rate als möglicher alleiniger Steuerungsparameter für den Sperrbereich bei auslastungsbezogener Erkennung
- Einführung der CNFL-Rate als zusätzlicher Sperr-Indikator bei der MPL-Erhöhung
- Parametrisierung der Korridorparameter bei der BOT-WS-Engpaßbehebung für den CPU- und Sperrbereich
- Einführung des Performance-Index als zusätzlichen Steuerungsparameter
- Modifikation der Cut-Regel bei der MPL-Verminderung
- Korrektur der Deadlock-Erkennung

4.5.1 Veränderungen in der topology-Datei

Die auslastungsbezogene Erkennung kann durch Setzen der Flag-Variablen `NO_TIME_CHECK_BN` eingeschaltet werden. Das System verwendet dann keine Antwortzeiteile mehr für die Engpaßerkennung und Analyse.

Die Verwendung der WKM-Rate als Indikator für einen Sperrengpaß kann durch das Setzen des Flags `USE_WEIKUM` eingestellt werden. An Stelle der CNFL-Rate wird dann anschließend die WKM-Rate verwendet.

Die Parameter für die Bestimmung des Sicherheitskorridors bei der BOT-WS-Engpaßbehebung wurden parametrisiert. Die Konstanten `N_HIGH_CPU_THRSH` und `N_HIGH_CNTNTN_THRSH` wurden in Variablen umgewandelt und werden jetzt durch die `topology-Datei` initialisiert.

Die Verwendung des Performance-Index zur Prioritätsbestimmung wird im `topology-Bereich`

für den TM-Modul eingeschaltet. Durch das Setzen des Flags `USE_PL` wird der TM-Modul veranlaßt, die dynamische Prioritätsberechnung vorzunehmen.

4.5.2 Veränderungen im Lastkontrollmodul

Die Veränderungen im Lastkontrollmodul bestanden neben dem Setzen der o.g. Flag-Variablen vor allem in der Modifikation der Cut-Regel und in der Veränderung der Verwendung der Antwortzeit bei der CPU-Engpaßerkennung.

Die Cut-Regel erwies sich in den Simulationen als zu radikal. Bei der Variablen `Local_Load_Monitor^TM_util` handelt es sich um die momentane Auslastung des MPL-Wertes. Die Berechnungsformel lautete in ihrer ursprünglichen Form:

$$\text{diff_util} := (1.0 - \text{Local_Load_Monitor}^{\wedge}\text{TM_util} \text{ for_Type } I);$$

Diese Formel wurde etwas abgeschwächt, indem die Differenz halbiert wird:

$$\text{diff_util} := (1.0 - \text{Local_Load_Monitor}^{\wedge}\text{TM_util} \text{ for_Type } I) / 2.0;$$

Bei der Erkennung eines CPU-Engpasses wird nicht mehr die gesamte CPU-Zeit verwendet (vgl. Tabelle 4.1). Es wird nur noch die CPU-Wait-Time für die Analyse verwendet. Dieses Vorgehen erschien gerechtfertigt, da die CPU nicht preemptiv verwaltet wird. Die maßgebliche Größe für den Engpaß ist somit die reine Wartezeit auf die CPU.

4.5.3 Veränderungen im CC-Modul

Bei der Durchführung der Simulationen wurde ein Problem bei der Deadlockerkennung entdeckt. Die implementierte Methode war nicht in der Lage, alle Deadlocks zu erkennen. Das Problem konnte durch weitere Simulation so weit bestimmt werden, daß sich folgende Situation darstellte:

Zu einem bestimmten Zeitpunkt entstand ein Deadlock durch den Zyklus (TA-IDs)¹:

372 -> 599 -> 514 -> 663 -> 125 -> 372

Der Ursprung dieses Zyklus findet sich einem früheren Zeitpunkt. Zu diesem Zeitpunkt gibt die TA 142 ihre Sperren aufgrund eines Abbruchs auf. Die wartenden Transaktionen sind

259 (R)

854 (W) (wird sofort nach Sperranforderung abgebrochen,
da einDeadlock erkannt wurde)

125 (R)

Durch den Abbruch der TA 854 wird die TA 125 auch zum Sperrinhaber, da es sich um kompatible Zugriffsmodi handelt. Der Wartegraph des Deadlock-Moduls wurde jedoch nicht verändert. Die Wartebезziehung 125->259 blieb bestehen, obwohl die TA weitere Sperranforderungen vornehmen kann (TA 125 fordert die Sperre auf ein Objekt an, für das TA 372 Sperrbesitzer ist).

Das Verfahren wurde dahingehend abgeändert, daß bei Vergabe der Sperren aufgrund kompakter Sperren, die Wartebезziehung zwischen altem und neuem (zusätzlichen) Sperrinhabern gelöscht wird und alle Wartebезziehungen, die auf den alten Sperrinhaber verweisen, auch auf den neuen Sperrinhaber zeigen. Im o.a. Beispiel war die Kante von 663 auf 125 nicht eingetragen worden. Durch das Übertragen aller eingehenden Kanten von 259 auch auf 125 wird die Zyklusbildung ausgeschlossen.

1. Der Zyklus wurde nicht aus dem Deadlock-Wait-Graph entnommen, sondern aus der lokalen Sperrtabelle abgeleitet.

5. Lastkontrolle bei CPU-Engpässen

Durch den Wert des MPL wird bestimmt, wie viele Transaktionen sich zum gleichen Zeitpunkt im System befinden dürfen. Die Untersuchungen im Bereich der CPU-Engpaßregelung müssen sich an bestimmten Grundfragen orientieren. An erster Stelle steht dabei die Frage nach der Existenz eines optimalen MPL-Wertes für eine bestimmte Last oder auch für alle Lasten. Simulativ wird deshalb bestimmt, welcher statische MPL-Wert die besten Ergebnisse liefert. Ausgehend von einem so bestimmten statischen MPL-Wert und den dazugehörigen Ergebnissen, ist dann zu ermitteln, ob die dynamische Regelung diese Ergebnisse erreicht oder sogar übertreffen kann. Hand in Hand mit dieser Aufgabe geht die Frage nach der Unabhängigkeit der Ergebnisse vom MPL-Startwert und die Frage nach der optimalen Justierung der internen Parameter (Antwortzeit-Prozentanteile und die CPU-Auslastung). Zur Bewältigung dieses Aufgabenkomplexes konzentrieren sich die Analysen in der ersten Phase auf die Betrachtung eines einzelnen Transaktionstyps.

Der zweite Aufgabenkomplex gewinnt erst bei mehreren Transaktionstypen an Bedeutung. Wie soll das System die knappe Ressource konkurrierenden Transaktionsklassen zuordnen. In diesen Bereich fallen die Analyse der LMPL-Regelung und die Betrachtung von Prioritätsalgorithmen für die BOT-WS und CPU-Queue.

Der CPU-Engpaß gehört zu der Gruppe der primären Engpässe. Die Erkennung eines CPU-Engpasses führt immer zu Verminderung des MPL-Wertes. Eine Verringerung des MPL-Wertes kann einen längeren Verbleib der Transaktionen in der BOT-WS bewirken. Die Simulationen eines CPU-Engpasses bedeutet somit in allen relevanten Fällen eine Betrachtung des Spannungsfeldes BOT-WS-CPU-Engpaß. Es ist somit nicht allgemein klar, ob eine Verringerung des MPL-Wertes überhaupt notwendig ist. Es stellt sich die Frage, ob es nicht besser ist, allen Transaktionen den Zutritt ins Verarbeitungssystem (d.h. ihre BOT-Behandlung zuzulassen) zu gewähren. Die maximale Auslastung der CPU liegen bei 100%. Bei dieser Auslastung wird ein Durchsatzmaximum für die Instruktionsverarbeitung erreicht. Da auf allen Stufen CPU-Requests ausgelöst werden (vgl. Abb. 4.3), ist es unwahrscheinlich, daß durch die „Flutung“ des Systems mit allen ankommenden Transaktion noch weitere Engpässe entstehen. Die Kapazität der CPU kann auf diese Weise vollständig abgeschöpft werden. Antwort auf diese generelle Evaluierung findet sich in den Simulationsergebnissen.

Um eine permanentes Schwingen des MPL-Wertes um einen optimalen MPL-Wert zu verhindern, wurde die Erhöhung des MPL-Wertes bei Erkennung eines BOT-WS-Engpasses an eine Bedingung gebunden. Die Lastkontrollkomponente überprüft die CPU-Auslastungsgröße. Der Wert für die maximal erlaubte CPU-Auslastung wird in den Simulationsläufen vorgegeben. Dieser Wert wird vor der Prüfung um einen vorgegeben Prozentsatz verringert. So wird bei einer erlaubten Auslastung von 95% und eines 5% Sicherheitsintervalls bzw. Korridors, der MPL-Wert nur dann erhöht, wenn die momentane CPU-Auslastung unter 90% liegt. Bei der antwortzeitbezogenen Engpaßerkennung wird zusätzlich zum Auslastungsindikator der CPU-Wait-Bestandteil überprüft. Die Erhöhung wird abgelehnt, sobald eine der beiden Bedingungen zutrifft.

Die CPU wird durch Anforderungen von verschiedenen Komponenten belastet. Die Tabelle 5.3 zeigt die für die einzelnen Komponenten angesetzten CPU-Instruktionen. Des weiteren werden in der Tabelle verschiedene für die Simulation verwendeten Parameter aufgeführt.

Die mittlere Anzahl der CPU-Instruktionen für die E/A wurde auf 3000 Instruktionen festgelegt. In den Graphiken wird als alternativer Begriff für die E/A auch an einigen Stellen Fix-Aufruf oder einfach nur FIX verwendet. Die 3000 Instruktionen für den CPU-Vorlauf fallen nur dann an, wenn die Seite nicht im Puffer gefunden wurde.

Eine weiterer Faktor, der zur Belastung der CPU beiträgt, ist die Lastkontrollkomponente. Die Kosten für die Bearbeitung der Kontrolle wurden auf einen Wert von 50.000 Instruktionen gesetzt. Dieser Wert wird noch einmal um 50.000 Instruktionen erhöht, falls die Komponente den Versuch unternimmt, einen erkannten Engpaß zu beheben. Diese Kosten treten in jedem Regelungsintervall auf. In den Simulationen wurde immer ein Regelungsintervall von einer Sekunde angenommen. Die zusätzliche CPU-Belastung durch die Lastkontrolle liegt für die angesetzten 5 MIPS zwischen 2 und 4 Prozent der Kapazität.

Die peripheren Speicher werden durch ein Log- und ein DB-Plattenmodul simuliert. Die DB-Platten haben eine mittlere Zugriffszeit von 15 ms und eine Übertragungszeit von 400 us. Die Log-Platten haben eine Zugriffszeit von 5 ms und eine Übertragungszeit von 400 us. Für die Log-Platten wurde ein nicht-flüchtiger Cache-Speicher mit einer Kapazität von 500 Seiten ein-

gestellt. Die Anzahl der Controller der DB- und Log-Platten wurde auf jeweils 200 gesetzt, um Engpässe bei der Controller-Belegung auszuschliessen.

Parameter	Einstellungen
Partition	
Anzahl	2
Anzahl Objekte	100.000-1.000
Blockungsfaktor	10
Spergranulat	Objekt
Spermethode	2-phasesiges Sperren
Puffergröße	2.000 Seiten
CPU	
Anzahl Prozessoren	2
Leistung (je CPU)	2,5 MIPS
Instruktionen für	
BOT	25.000
EOT	25.000
Referenz	5.000
I/O	3.000
Lastkontrolle (Instr./s)	50.000 / 100.000
Peripherie	
Zugriffszeit (Log-Platte)	5 ms
Zugriffszeit (DB-Platte)	15 ms
Übertragungszeit	400 ms
Anzahl Controller	200

Tabelle 5.3: Parameter der CPU-Engpaß-Simulationen

5.1 Simulationen mit stabiler Last

Ausgangspunkt der Untersuchung ist die Betrachtung von Transaktionen eines einzelnen Transaktionstyps im System. Die anfängliche Konzentration auf nur einen Typ bringt es mit sich, daß in diesem Stadium nur die statische MPL-Wert Einstellung (SMPL) und das GMPL-Verfahren von Relevanz sind. Die gewonnenen Ergebnisse bilden die Grundlage für die Analyse des lokalen und globalen Verfahrens in weiteren Untersuchungen.

In den Simulationsstudien mit nur einem Transaktionstyp wurde die Konfiguration wie folgt gewählt: Eine Transaktion referenziert 40 Objekte und benötigt somit ca. 370.000 Instruktionen ($40 \cdot (5000 + 3.000) + 50.000$). Diese Instruktionsanzahl beinhaltet auch die E/A-CPU-Vorläufe. Es handelt sich um eine hundertprozentige Leselast. Diese Annahme umgeht mögliche Engpässe auf der Sperr- und E/A-Ebene. Die Puffergröße beträgt 2.000 Seiten. Ausgehend von einer Gesamtseitenzahl von 10.100 ergibt sich ein Verhältnis DB-Größe zu Seitenpuffer von 1:5. Dies schlägt sich für die gegebene Last und Referenzverteilung in einer Puffertrefferrate von ca. 40% nieder. Folglich sind für eine Transaktion im Mittel lediglich 24 Plattenzugriffe notwendig. Die Anzahl der benötigten Instruktionen sinkt somit auf 322.000 Instruktionen ($40 \cdot (5000 + (0.60 \cdot 3000)) + 100.000$).

Die Prozessorleistung des Systems wurde relativ knapp dimensioniert, um einen CPU-Engpaß zu ermöglichen. Es werden zwei Prozessoren mit jeweils 2,5 MIPS simuliert. Als Zielvorgabe wurde eine Antwortzeit von 1,5 s angesetzt. Die Antwortzeit-Bestandteile wurden auf 25% für die CPU-Wartezeit und 10% für die BOT Wartezeit gesetzt. In der Tabelle 5.3 sind die relevanten Parameter noch einmal übersichtsartig dargestellt.

5.1.1 Variation der Ankunftsrate und des MPL-Startwertes

Die Simulationen mit stabilen Ankunftsrate wurden mit einer Reihe der systematischen Variation der Ankunftsrate begonnen. Die Ankunftsrate sind innerhalb einer einzelnen Simulation nicht variiert worden. Die Simulationen wurden mit verschiedenen Anfangskonfigurationen gestartet. Die Ankunftsrate von 12 bis 15 tps stellen unterschiedliche Belastungsprofile dar. Die rechnerische Belastung liegt ohne Berücksichtigung der Lastkontrolle bei Ankunfts-

rate von 12 tps bei 3,9 MIPS respektive bei 78 % der Kapazität. Bei einer Ankunftsrate von 15 tps liegt die Belastung bei 4,8 MIPS bzw. 96 % der Kapazität.

5.1.2 Antwortzeit-Aufgliederung bei statischer MPL Einstellung

Für den statischen Fall wurde in weiteren Studien der MPL-Wert 20 als optimale Konfiguration für die betrachteten Lasten ermittelt.

Bei den Simulationen trat jedoch ein Problem auf: da jede einzelne Referenz einer Transaktion auch einen CPU-Vorlauf für die E/A beinhaltet, wirkte sich ein zu hoher MPL-Wert negativ auf die E/A-Bearbeitung aus. Aus diesem Grund erzielte auch der MPL-Wert von 20 ein besseres Ergebnis als ein MPL-Wert von 50, obwohl sich für letzteren eine quasi 100-prozentige Auslastung der CPU ergibt (vgl. auch Abb. 5.1). Die Antwortzeit für den CPU-E/A-Vorlauf mußte bei enger Auslegung zur CPU-Bearbeitungszeit der Transaktionen gerechnet werden. Die eigentlichen Verzögerungen betreffen ausschließlich die CPU und nicht die E/A. Die benutzte Statistik kann also rein theoretisch dazu führen, daß bei antwortzeitbezogener Erkennung ein E/A-Engpaß erkannt wird, da der Plattenengpaß vor dem CPU-Engpaß geprüft wird. Dies wirkt sich in den Simulationen aus zwei Gründen nicht aus. Der Antwortzeitanteil für den Plattenengpaß wurde sehr hoch gewählt und wird bei dem verwendeten Szenario nicht erreicht. Der andere Grund besteht in der Regelung selbst. Bei einer hier unterstellten falschen Erkennung eines eigentlichen CPU-Engpässes als Plattenengpaß wird nicht unterschiedlich vorgegangen: Es wird der MPL-Wert reduziert.

Um diesen negativen Effekt bei der E/A, der nur bei sehr hoher CPU-Auslastung auftritt, entgegen zu wirken, wurden bei den dynamischen MPL-Varianten mit restriktiveren CPU-Auslastungsprozentsätzen als 100% gearbeitet¹.

1. Die Anpassung der Statistik ist für die nächste Version vorgesehen. Der Antwortzeitanteil des CPU-E/A-Vorlaufes, wird dann beim Antwortzeitanteil der CPU-Bearbeitung eingerechnet.

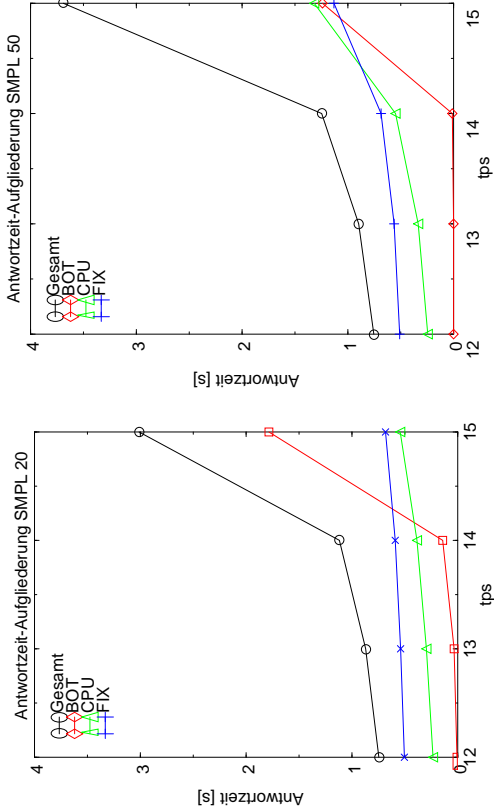


Abb. 5.1: Antwortzeit-Ergebnisse für SMPL 20 und 50

5.1.3 Ergebnisse der globalen MPL-Regelung

Abb. 5.2 bis 5.4 zeigen die für drei dynamische und die statische Variante(n) gewonnenen Resultate. Für die MPL-Startwerte (im folgenden MPL[0] genannt) 1, 10 und 100 stellen die Grafiken die Antwortzeiten und den Anteil der Transaktionen, die ihre Zielantwortzeit (goal) einhalten (TaInTime-Anteile), dar. Es fällt sofort ins Auge, daß die Antwortzeiten stark vom maximal erlaubten Auslastungsgrad und vom MPL[0]-Wert beeinflusst werden. Je größer der Begrenzungswert ist, desto besser sind die Antwortzeiten und die TaInTime-Anteile. In Abb. 5.2 wurde zum Vergleich der statische MPL-Wert auf 10 gesetzt. Schlechter als der statische MPL-Wert von 10 schneidet nur die Regelung mit einem Grenzwert von 85 % ab. Bereits ab einer Ankunftsrate von 13 tps baut sich die BOT-Warteschlange auf. Mit Ausnahme des 85%- Laufes werden in allen dynamischen Varianten durchschnittliche MPL-Werte erreicht, die höher als zehn liegen. Die CPU wird somit besser genutzt und der Antwortzeit verbessert.

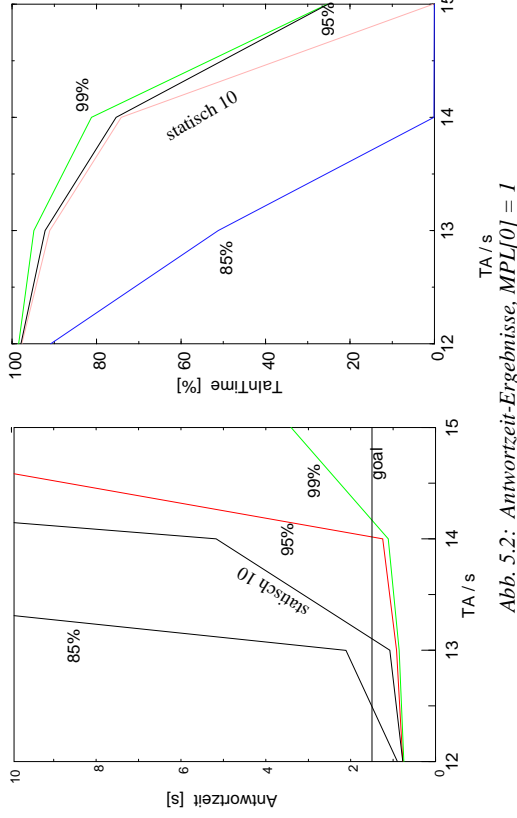


Abb. 5.2: Antwortzeit-Ergebnisse, MPL[0] = 1

In den beiden folgenden Simulationsläufen wurde zum Vergleich die SMPL-Kurve mit MPL-Wert 50 angegeben. Die Ergebnisse der dynamischen Verfahren nähern sich dabei bei steigenden MPL-Startwert immer näher an. Als Trend bleibt festzuhalten, daß der beste ermittelte statische MPL-Wert und das dynamische Verfahren mit einer 99% Grenze und einem 5% mit dem vorliegende Verfahren möglich, einem iterativ ermittelten besten statischen Wert ein Bezug auf das Ergebnis äquivalentes dynamisches Verfahren entgegenzustellen.

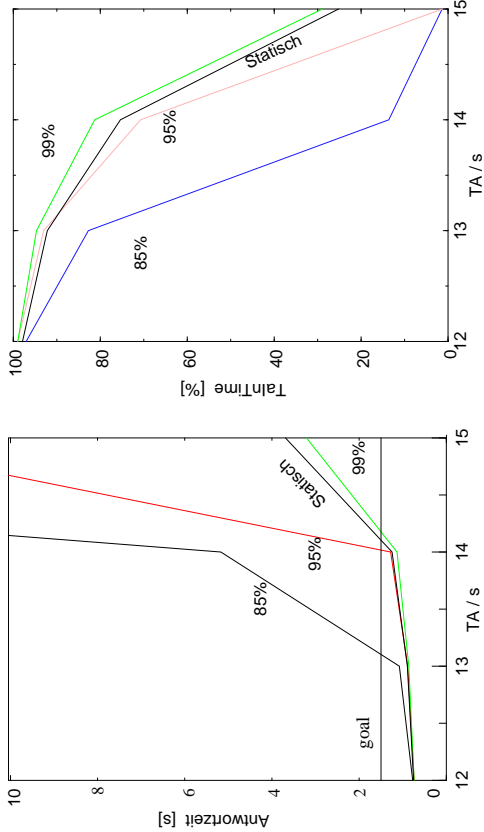


Abb. 5.3: Antwortzeit-Ergebnisse, $MPL[0] = 10$

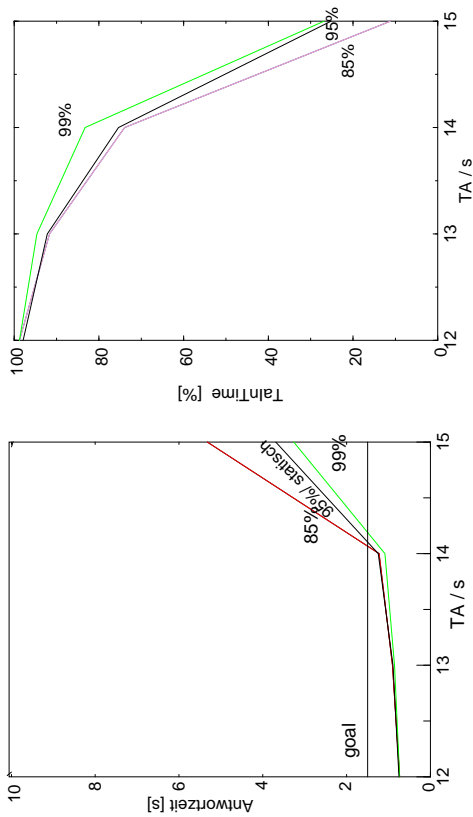


Abb. 5.4: Antwortzeit-Ergebnisse, $MPL[0] = 100$

Vergleicht man die drei Simulationsläufe mit verschiedenen Start-MPL-Werten, so fällt auf, daß die Ergebnisse durchaus unterschiedlich ausfallen. Die Unterschiede zwischen den Antwortzeiten, die mit MPL-Wert 1 und 10 starteten, unterscheiden sich lediglich marginal. Offensichtlich ist aber der Unterschied zwischen den ersten beiden Simulationsläufen und

dem Simulationslauf mit MPL-Startwert 100. Während sich bei den ersten beiden Läufen schon erhebliche Antwortzeitunterschiede ab einer Ankunftsrate von 13 tps zeigten, ist bei einem Startwert von 100 erst ein relevanter Unterschied bei einer Ankunftsrate von 15 tps zu erkennen. Die Annäherung des zeitabhängigen $MPL[t]$ -Wertes an den optimalen MPL_{opt} -Wert (MPL_{opt}) ist somit abhängig von der Richtung, von der auf MPL_{opt} *zugewandert* wird. Wird von oben auf MPL_{opt} *zugewandert*, so erhält man in der Regel ein besseres Ergebnis als wenn dies von unten geschieht. Dies gilt aber nur, wenn nicht durch die Auslastungsregelung der MPL_{opt} unterschritten wurde. Grund für dieses Verhalten ist die Verwendung des Sicherheitsintervalls. In den Simulationsläufen wurde ein Sicherheitsintervall von 5% benutzt. Dies bedeutet z.B. für den 85%-Fall, daß bei einer *momentanen* CPU-Auslastung (bzgl. letztem Regelungsintervall gemittelter Wert) von 80% keine MPL-Erhöhung mehr vorgenommen wird. Dies resultiert in den gezeigten Fällen zumeist in einem geringeren MPL-Wert. In Tabelle 5.4 werden die mit den Verfahren durchschnittlich erhaltenen MPL-Werte aufgelistet. Die Daten verdeutlichen noch einmal den soeben dargelegten Sachverhalt. Die einzelnen Werte in einer Zelle geben die MPL-Werte an, die mit den verschiedenen Start-MPL-Werten (1-10-100) erzielt wurden.

Grenzparameter	12 tps	13 tps	14 tps	15 tps
85%	9-10-100	9-10-11	9-10-11	9-10-12
95%	9-10-100	11-11-12	11-11-12	11-11-14
99%	9-10-100	11-11-12	14-14-14	14-14-15

Tabelle 5.4: Durchschnittliche MPL-Werte Simulationslauf 1

Für die weitere Behandlung des CPU-Engpasses lassen sich folgende Zwischenergebnisse festhalten:

- (1) Die Begrenzung für das Sicherheitsintervall sollte möglichst hoch liegen. Eine vollständige Freigabe der Begrenzung führt jedoch zu einer überhöhten CPU-Wartezeit und sollte vermieden werden.
- (2) Der MPL-Startwert sollte generell über dem zu erwartenden MPL_{opt} liegen.
- (3) Das Sicherheitsintervall sollte zwar mit Vorsicht gewählt, aber dennoch knapp bemessen werden, um eine mgl. Unterschreitung des MPL_{opt} wieder ausgleichen zu können.

5.1.4 Variation des MPL- Startwertes für zwei Lastsituationen

Ein wesentliches Charakteristikum eines guten Regelungssystems besteht in der Unabhängigkeit der Regelung von den Anfangsparametern. Neben den internen Parametern für die Engpaßbestimmung und die Gegenmaßnahmen ist, wie im letzten Abschnitt bereits erörtert, der Anfangswert des MPL von Relevanz. Das System soll dabei unabhängig vom jeweils gewählten Anfangswert zu stabilen Resultaten kommen. Diese Bedingung ist notwendig um zum einem eine Unabhängigkeit von vollkommen abwegigen Werten zu schaffen und zum anderen, um eine gewisse Robustheit gegenüber extremen Konstellationen zu gewährleisten. In diesem Abschnitt wird der Start-MPL für zwei Lastsituation variiert. Der CPU-Auslastungsgrenzwert wurde auf 99 % mit 1 % Korridor gesetzt.

Aus den Reihen, die unter 5.1.1 beschrieben wurden, wurden die beiden Fälle einer statischen Last mit konstanter Ankunftsrate von 12 und 15 tps ausgewählt. Zum Vergleich wurden in die Grafiken auch die erhaltenen Resultate für einen konstanten MPL-Wert übernommen. Diese erfüllen dabei zwei Funktionen:

- (1) Sie zeigen die Ergebnisse, die man ohne Regelung des MPL-Wertes erhält.
- und
- (2) das oder die optimalen Ergebniss(e), die man überhaupt mit einem statischen Verfahren erzielen kann.

Die Ergebnisse aus Abb. 5.5 lassen sich wie folgt bewerten: Die Last von 12 tps stellt für das System eine Unterlastsituation dar. Der MPL-Wert kann für diesen Fall lediglich zu gering gewählt werden. Die Wahl eines hohen MPL-Wertes wirkt sich folglich nicht in einer andersonigen Betriebsmittelrestriktion aus. Es zeigt sich jedoch, daß zumindest ein Wert von 10 gewählt werden muß. Das dynamische Verfahren ist in allen Bereichen genau so gut wie das statische. In den Anfangsbereichen ist es dem statischen sogar überlegen, da der zu niedrige MPL-Wert erkannt und korrigiert wird. Der Overhead für die Lastregelung wirkt sich nicht auf die Leistungsfähigkeit des Systems aus, da die Regelung nur am Anfang eingreifen muß und die CPU durch die gesamte Last auch nicht vollständig in Anspruch genommen wird.

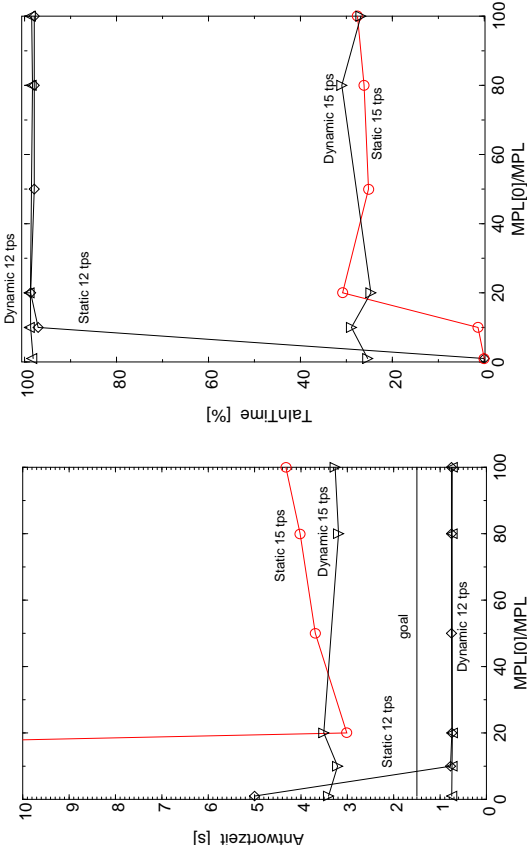


Abb. 5.5: Antwortzeitverhalten Simulation 1.1

Die Situation stellt sich im 15-tps-Fall anders dar. Es handelt sich hierbei um eine Hochlast. Die Auslastung der CPU liegt im Durchschnitt der Simulationen bei ca. 98 %. Für einen MPL-Wert von 20 liefert das statische Verfahren einen Antwortzeit-Wert, der unter dem des dynamischen Verfahrens liegt. Die Werte der über den gesamten Simulationszeitraum gemittelten Werte des MPL werden in Tabelle 5.5 aufgelistet.

Tabelle 5.5: Durchschnittliche MPL-Werte bei Ankunftsrate 15 tps

Startwert	1	10	50	100
Durchschnittlicher MPL-Wert	13	14	14	15

Das gute Abschneiden des statischen Verfahrens ist jedoch zu relativieren:

- (1) Die Situation stellt eine Überlast bzgl. der Zielantwortzeit dar. Dies führt bei dem dynamischen Verfahren zu quasi permanenten Regelungsmaßnahmen. Die CPU wird zusätzlich zu den 50.000 Instruktionen ständig mit weiteren 50.000 Instruktionen für die Engpaßbehebung belastet. Bei einer den tatsächlichen Verhältnissen angepaßten Zielvorgabe träte die Belastung somit in geringerem Maße auf.

- (2) Die statische MPL-Methode liefert nur für einen bestimmten Bereich ein optimales Ergebnis. Das dynamische Verfahren liegt zwar für keinen Startwert unter dem besten statischen Wert. Die Werte liegen jedoch nicht um Größenordnungen auseinander. Darüber hinaus erreicht das dynamische Verfahren für jeden Anfangswert ein Ergebnis, das max. 10% über dem des besten statischen Ergebnisses liegt.

5.2 Simulationen mit variabler Last

Um das Verhalten des Systems bei sich stark verändernden Lastprofilen zu untersuchen, wurden Simulationen mit mehreren Lastspitzen durchgeführt. Die Belastung wurde dabei so gewählt, daß während der Lastspitzen eine vollständige Überlastsituation entsteht. Die Last bestand wiederum aus nur einem Transaktionstyp. Die Grundankunftsrate lag bei 12 tps und die Spitzenankunftsrate bei 25 tps. Über den gesamten Simulationszeitraum (von 1000 s) wurden 5 Lastspitzen mit einer Dauer von jeweils 15 ms verteilt. In den 15 ms andauernden Lastspitzen wird das System mit einer Last konfrontiert, die bei ca. 165% der CPU-Kapazität liegt. Die Resultate dieser Simulation werden in Abb. 5.6 gezeigt.

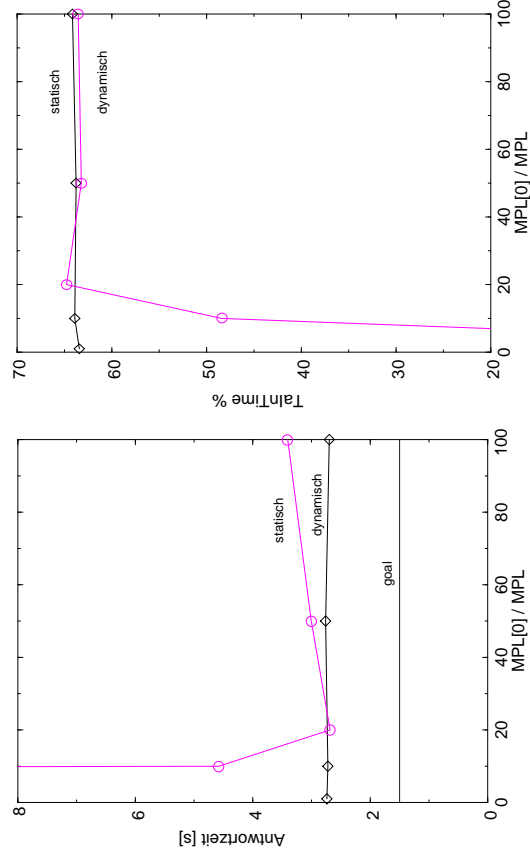


Abb. 5.6: CPU-Engpaß und variable Last

In Abb. 5.6 werden die Verläufe der Antwortzeit und der Prozentsatz der die vorgegebene Antwortzeit einhaltenden Transaktionen dargestellt. Es zeigt sich, daß das dynamische Verfahren in fast allen Bereichen besser abscheidet als die statische Variante. Lediglich für den statischen MPL-Wert von 20 wird sowohl die Antwortzeit als auch die TaktTime-Anteile des dynamischen Verfahrens leicht unterschritten bzw. übertroffen. Im Vergleich zu den Simulationen im Abschnitt 5.1.1 kann das Verfahren hier auch als stabil bzgl. der MPL-Anfangskonfiguration bezeichnet werden.

Die Abb. 5.7 zeigt den Verlauf der Antwortzeit und die korrespondierende MPL-Entwicklung für einen MPL-Startwert von 100. Es zeigt sich hier anschaulich, wie die Überlastphasen auf den Antwortzeitverlauf wirken. Die MPL-Entwicklung zeigt, wie sich das System flexibel bis an die Grenzen der Belastbarkeit hin ausbaut. Der $MPL[0]$ -Wert beträgt 100. Aus der Abb. 5.7 ist ersichtlich, wie durch die Cut-Regel der sehr hohe Anfangswert beim ersten Auftreten einer Zielverfehlung (Überschreitung des goals) sehr schnell abgesenkt wird. Die ersten Antwortzeitverfehlungen ereignen sich im Zeitpunkt 100 s. Der MPL-Wert wird innerhalb von 3 s von 100 auf 18 reduziert. Danach verläuft die MPL-Entwicklung stabil um einen durchschnittlichen MPL-Wert von 15. Die Sicherung bei der MPL-Erhöhung durch die Erkennung aufgrund von Antwortzeitanteilen und des CPU-Auslastungsindikatoren führen lediglich zu geringen Veränderungen des MPL-Wertes. .

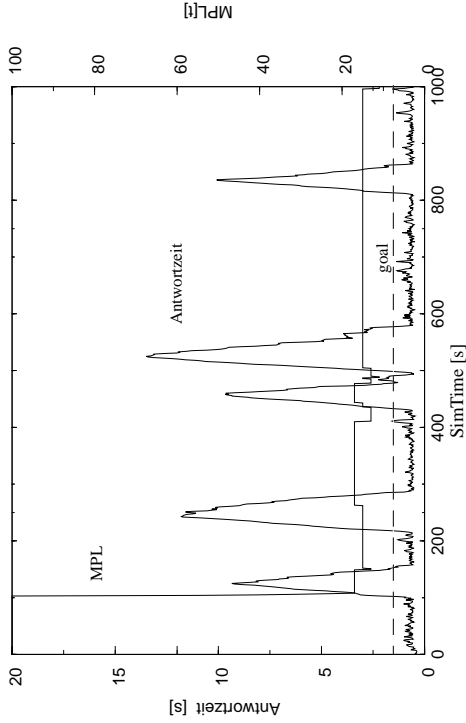


Abb. 5.7: Antwortzeit, Dynamic Load, $MPL[0] = 100$

Die Abb. 5.7 macht jedoch auch klar, mit welchen Einschränkungen dieses Ergebnis gesehen werden muß. Die Dauer der Hochlastphase beträgt insgesamt lediglich 75 ms. Dies ist jedoch bzgl. des Simulationszeitraums ein sehr kleines Zeitintervall. Obwohl diese kurzen Überlastphasen eine längere Ausschwingphase der Antwortzeitkurve bewirken, beträgt die Phase, in der die Antwortzeit verfehlt wird, weniger als ca. 30% des gesamten Simulationszeitraumes.

Nichtsdestotrotz zeigt sich an dem repräsentativ ausgewählten Simulationsbeispiel, daß die mit dem dynamischen Verfahren erhaltenen Ergebnisse auch und insbesondere für eine dynamische Last ein gutes Ergebnis liefern.

5.3 Sensitivitätsanalyse

In den letzten Abschnitten wurden die internen Simulationsparameter der Antwortzeitanteile und des CPU-Auslastungsindikators für die Lastkontrolle zumeist als gegeben vorausgesetzt. Die Wahl der Werte für diese Parameter basierten auf reinen Plausibilitätsüberlegungen und gewissen Erfahrungswerten. Es sind somit zwei Aspekte von Interesse:

- Für welche internen Parameter werden die besten Ergebnisse erzielt.
- Wie genau muß der beste interne Parameter gewählt werden bzw. wie groß ist das Intervall der internen Parameter, die ein ähnlich gutes Ergebnis, wie der beste erzielen.

Zur Untersuchung dieses Fragestellungen wurden vier weitere Simulationsreihen angelegt. Es wurden die Fälle mit einer Ankunftsrate von 14 und 15 tps gewählt. Es werden in diesen Reihen die Grenzwerte für die Antwortzeitanteile der CPU- und BOT-Engpaßerkennung variiert. Zusätzlich wurde eine Reihe für den Fall mit variabler Last angelegt. Als letzte Untersuchung wird ein Versuch dargestellt, in dem die Regelung allein über die CPU-Auslastungsindikatoren gesteuert wird.

5.3.1 Variation der Antwortzeitanteile

Die Lastkontroll-Komponente analysiert die Antwortzeitanteile der im Beobachtungsintervall beendeten Transaktionen. Gemäß der im Abschnitt 3 beschriebenen Erkennungshierarchie wird erst auf CPU-Engpaß und dann auf BOT-Engpaß geprüft. Die Schwellwerte für den Analysevektor wurden für den CPU-Anteil auf 25% und für den BOT-Anteil auf 10% gesetzt. Falls kein anderer Engpaß vorhanden ist, wird durch die Wahl des BOT-Wertes die Häufigkeit der MPL-Erhöhungsversuche geregelt. Bei der CPU-Auslastung handelt es sich um einen der Ankunftsrate entsprechenden statistisch schwankenden Wert. Wird in den einzelnen Regelszeitpunkten nur ein BOT-WS-Engpaß erkannt, so wird eine Erhöhung nur vorgenommen, wenn der CPU-Wait-Anteil und die CPU-Auslastung nicht zu hoch (z.B. hier unter 98 % liegt). Geht man nun davon aus, daß kein CPU-Engpaß erkannt wird und die CPU-Auslastung

in einigen Regelungszeitpunkten unter die 98% Grenze sinkt. Eine MPL-Erhöhung findet dann jedoch nur statt, wenn in diesen Zeitpunkten auch ein entsprechender Engpaß erkannt wird. Legt man somit den BOT-WS-Engpaß-Anteil auf eine relativ geringen Wert (z.B. 5%) fest, so ist die Bedingung für den BOT-WS-Engpaß sehr häufig erfüllt und selbst bei sporadisch auftretender Unterschreitung des Grenzwertes von 98% wird der MPL-Wert erhöht. Diese Wahrscheinlichkeit nimmt bei Vergrößerung des BOT-WS-Anteils aus offensichtlichen Gründen immer weiter ab.

Die Abb. 5.8 und 5.9 zeigen den Verlauf der Antwortzeit bei veränderten Parametern für die CPU- und BOT-Antwortzeitanteile. In den Kurven werden beide Parameter gleichgewichtet variiert und auch jeder Parameter einzeln bei konstant gehaltenem Alternativwert. Derjenige Parameter, der nicht variiert wird, nimmt den konstanten Wert von 25% an.

Bei einer Last von 14 tps ist der BOT-Anteil unkritisch. Die Antwortzeit reagiert nur auf den CPU-Anteil. Die Variation des BOT-Anteil bei konstantem CPU-Anteilswert von 25% ergibt quasi eine zur X-Achse parallele Gerade. Gute Ergebnisse werden für CPU-Anteile von 25% bis 40% der Antwortzeit erzielt. Die Quote für den TaInTime-Anteil zeigt jedoch, daß der optimale Wert hier bei 25% liegt. Dieses Ergebnis wird noch dadurch unterstrichen, daß die BOT-Variationskurve die obere Grenze für die maximal erzielbare TaInTime-Quote ist.

Auch in der Simulation mit 15 tps bestätigt sich das Ergebnis. Wie man aus Abb. 5.9 ersehen kann, ist sowohl in Bezug auf die Antwortzeit als auch bzgl. der TaInTime-Anteile der CPU-Anteilswert von 25% der optimale Wert. Obwohl die durchschnittliche Antwortzeit für alle Werte über der vorgebenen Antwortzeit liegt, erreichen dennoch ca. 30 % der Transaktionen ihr Ziel. Die gleichförmige Variation liefert bis zum Wert von 25% wesentlich schlechterer Antwortzeit-Ergebnisse als die beiden anderen Reihen. Die Unterschiede zwischen den Ergebnissen der Variation beider Anteile und nur des CPU-Anteils (bei 25% BOT-Einstellung) liegen in den wesentlich öfter auftretenden Regelungen bei ersterem. Die Einstellungsparameter von 5 % bis 10% sind zu niedrig für ein stabil verlaufendes Verfahren. Der MPL-Verlauf der CPU-Kurve ist wesentlich stabiler, da ein BOT-Engpaß bzw. dessen Erkennung relativ selten auftritt. Das beste Ergebnis liefert eine Einstellung von 25% für den CPU-Anteil. Bei dieser Einstellung kommt es außerhalb des anfänglichen Einschwingintervalls nur sehr selten zu einer MPL-Erhöhung. Der durchschnittlicher MPL-Wert liegt bei dieser Konfiguration durchgehend (d.h. für alle Einstellungen des BOT-Anteils) bei 16.

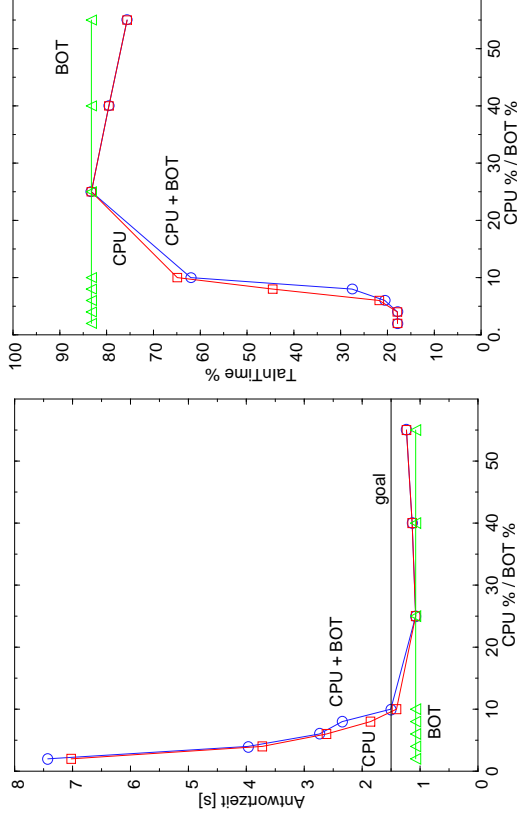


Abb. 5.8: Sensitivitätsanalyse, CPU-Engpaß, Ankunftsrate 14 tps

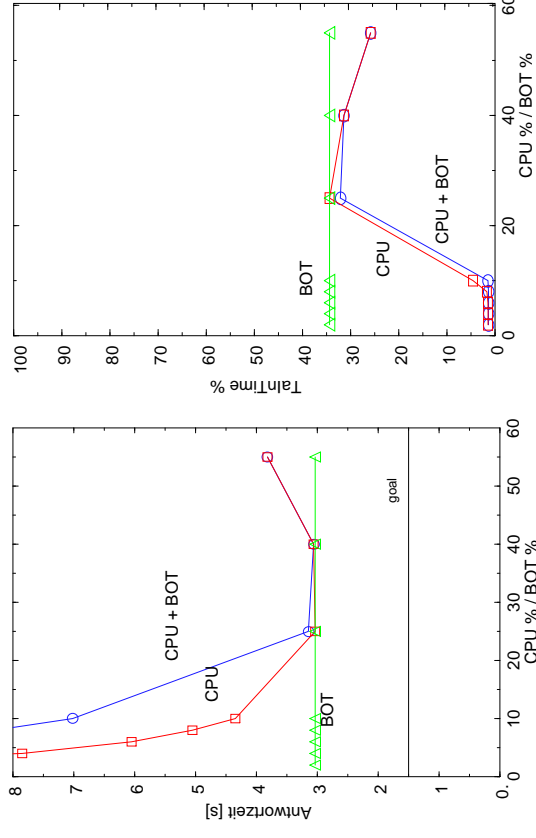


Abb. 5.9: Sensitivitätsanalyse, CPU-Engpaß, Ankunftsrate 15 tps

5.3.2 Regelung über den CPU-Auslastungsindikator

Die Erkennung des Engpasses wurde bisher immer über die Analyse der Antwortzeitanteile geregelt. Es besteht aber auch die Möglichkeit, die Regelung nur über den CPU-Auslastungsindikator zu steuern. Der CPU-Engpaß wird erkannt, wenn die CPU-Auslastung den vorgegebenen Grenzwert überschreitet. Ein BOT-WS-Engpaß wird nur erkannt, wenn die absolute Anzahl der Transaktionen in der BOT-Warteschlange größer als der eingestellte Grenzwert wird. Die Erkennung wird auch hier nur dann gestartet, wenn die mittlere Antwortzeit der Transaktionen ihre Zielvorgaben überschreitet.

In den Simulationen wurde die Auslastungsgrenze auf 99% der CPU-Kapazität gesetzt. Das Sicherheitsintervall wurde mit 1% sehr eng gewählt. Eine MPL-Erhöhung findet somit nur dann statt, wenn die momentane Auslastung geringer als 98% ist.

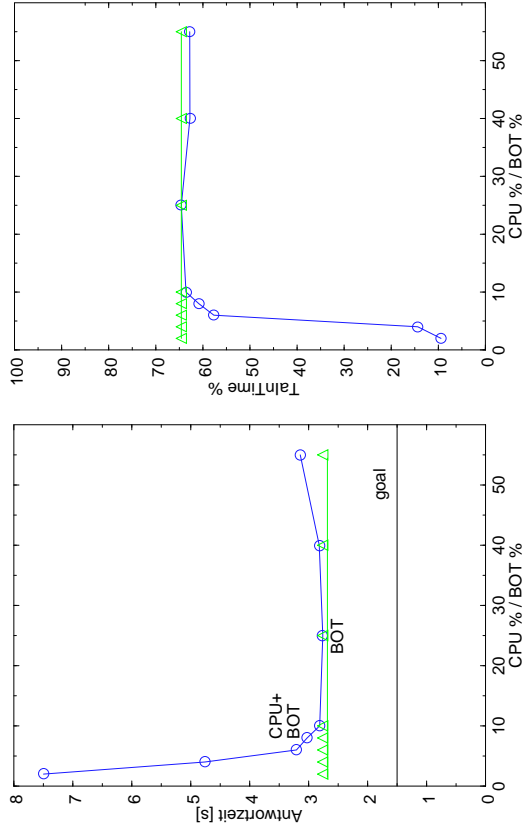


Abb. 5.10: Sensitivitätsanalyse, CPU-Engpaß, Variierende Last

Die Ergebnisse für die variierende Last weichen etwas ab. Der konstante CPU-Anteilswert von 25% stellt wieder die Untergrenze für die Antwortzeit und die Obergrenze für die TaktTime-Anteile auf. Die Kurven für den variierten CPU-Anteil und die Variation beider Begrenzungs-werte fallen hier jedoch zusammen. Für die betrachteten Werte ist die Einstellung des BOT-Grenzwertes soweit invariant bzgl. des Ergebnisses. Es entscheidet lediglich die CPU-Begren-zung. Die besten Ergebnisse lassen sich für eine CPU-Antwortzeit-Prozentwert von 10% bis maximal 25% erzielen. Die BOT-Variation zeigt aber auch hier, daß der CPU-Wert von 25% ein optimales Ergebnis liefert.

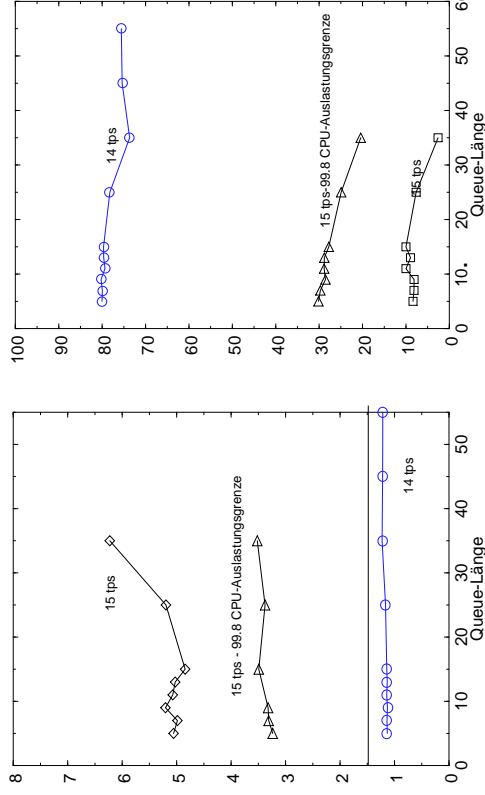


Abb. 5.11: CPU-Auslastungsbezogene Regelung mit 99% (14 u. 15 tps) und 99,8 Begrenzung (nur 15tps)

In Abb. 5.11 werden die Ergebnisse der diesbzgl. Simulation wiedergegeben. Die Werte für 14 tps entsprechen den Ergebnissen bei antwortzeitbezogener Erkennung. Für den 15-tps-Fall weicht die Antwortzeit im Vergleich jedoch erheblich ab. Bei der antwortzeitbezogenen Erkennung wird der MPL-Wert nur verringert, wenn der CPU-Antwortzeit-Anteil überschrit-ten ist. Bei der hier gewählten Regelung geschieht dies aber rein über die Auslastungs-

größe. Diese Vorgehensweise sorgt dafür, daß der Grenzwert ein **harter** Grenzwert ist. Die antwortzeitbezogene Erkennung steuert etwas ungenauer, da die Erkennung des CPU-Engpasses über einen abgeleiteten Wert erfolgt. Die Regelung mit der Auslastungsgröße drängt aber auf die absolute Einhaltung des Grenzwertes.

Ein Simulationslauf mit einer erhöhten CPU-Auslastungsgrenze von 99,8% der gesamten Kapazität und einem geringeren Sicherheitsintervall von 0.5 % verbesserten die Antwortzeit für den 15tps-Fall jedoch erheblich (vgl. Verlauf der 15tps-99,8 Kurve in Abb. 5.11). Die Antwortzeit sinkt dennoch nicht unter 3s. Dieses Ergebnis liegt noch über dem besten statischen Wert (vgl. Abb. 5.5), bei dem die Antwortzeit unter 3 s sinkt. Es liegt aber auf dem gleichen Niveau wie die anderen dynamischen Verfahren.

Die Erkennung eines BOT-WS-Engpasses wird über die **Länge der Warteschlange** gesteuert. Die Variation der BOT-Warteschlangenlänge als interner Steuerungsparameter zeigt lediglich beim 15-tps-Fall mit 99% Regelung und in abgeschwächter Form bei einer 99,8 % Grenze eine Auswirkung. Aus den Ergebnissen läßt sich ableiten, daß der BOT-Warteschlangenwert gering gehalten werden sollte. Bei zu großer Relaxierung dieses Wertes (z.B. 25 oder 30) bleibt der durchschnittliche MPL-Wert zu niedrig.

Auf die Darstellung der Ergebnisse der auslastungsbezogenen Engpasserkennung für die **variable Last** wird verzichtet, da die Regelung mit einer 99 % Grenze und einem 1 % Korridor ähnlich gute Ergebnisse wie das antwortzeitbezogene Verfahren liefert.

5.4 Simulationen mit 2 Transaktionstypen

Die bisher durchgeführten Experimente beschäftigten sich nur mit einem Transaktionstyp. Die Wahl des MPL-Wertes wirkte sich somit immer auf dieselbe Transaktionsklasse aus. Die gewonnen Ergebnisse sollen nun mit einer Last, die aus zwei verschiedenen Transaktionsklassen besteht, verglichen werden. Die Experimente sollen Aufschluß geben über Fragen wie :

- Wirken sich unterschiedliche MPL-Werte in gleicher Weise auf die Antwortzeit der Transaktionen aus ?
- Gibt es einen oder mehrere optimale MPL-Werte in diesem Fall ?
- Ist die Regelungsvariante mit einem globalen MPL-Wert optimal ?

Die Transaktionen konkurrieren im 2-Typen-Fall um die Engpaßressource CPU. Die ersten Experimente wurden ohne Berücksichtigung von bestimmten Prioritäten vorgenommen. Die Anzahl der Prozessoren wird beibehalten. Die beiden CPUs besitzen jedoch in den Simulationen eine Kapazität von je 12,5 MIPS. Es wurden zwei Transaktionstypen mit 5 und 50 Referenzen (Refs) angenommen. Die Ankunftsrate wurde stabil auf 200/20 tps gesetzt. Legt man eine durchschnittliche Puffertrefferquote von 40 % zugrunde, so erhält man eine durchschnittliche CPU-Belastung von ca. 24,6 MIPS oder 98 % der CPU-Kapazität. Die Zielantwortzeiten wurden auf 0,1 s für Transaktionstyp 1 und 1,0 s für Transaktionstyp 2 gesetzt. Die übrigen Parameter werden in Tabelle 5.6 und 5.6 wiedergegeben.

Die Datenbank besteht aus zwei Partitionen mit 1.000 und 10.000 Seiten. Die Transaktionen der Transaktionsklasse eins referenzieren beide Partitionen mit gleicher Häufigkeit. Die Transaktionen der Klasse zwei referenzieren zu 10 % Partition eins und zu 90 % Partition zwei.

Parameter		Typ 1	Typ2
Transaktionen			
Referenzen (pro Transaktion)		5	50
Zielantwortzeit (s)		0,1	1,0
Ankunftsrate (tps)		200	20
Schreibanteil		0%	0%
Zugriffsverteilung (Partition 1 / Partition 2)		(0,5,0,5)	(0,1,0,9)

Tabelle 5.6: Parameter der Transaktionen

Parameter	Einstellungen
Partitionen	
Anzahl	2
Größe (Anzahl Objekte)	1.000, 100.000
Blockungsfaktor	10
Puffergröße	2.000
CPU	
Anzahl	2
Leistung	12,5 MIPS

Tabelle 5.7: Grundparameter der CPU-Engpaß-Simulationen mit 2 Typen

Abb. 5.12 zeigt die Antwortzeitzusammensetzung für die beiden Typen bei variiertem statischen MPL-Wert. Anhand der Grafiken läßt sich verfolgen, wie der Antwortzeitanteil für den Verbleib in der BOT-Warteschlange sich bei wachsendem MPL-Wert immer weiter verringert. Es findet eine Antwortzeitverschiebung vom sinkenden BOT-WS-Anteil hin zum wachsenden CPU-Anteil statt. Für den Transaktionstyp 1 bedeutet eine Vergrößerung des MPL-Wertes bis zum Wert von 200 eine stetige Antwortzeitverbesserung. Für den Transaktionstyp 1 ist somit eine Begrenzung des MPLs nicht notwendig, da keine negativen Auswirkungen eines zu hoch gewählten MPL-Wertes zu ersehen sind. Für den Transaktionstyp 2 ergibt sich jedoch eine anderes Bild. Eine Antwortzeitverbesserung läßt sich hier nur bis zum MPL-Wert von 40 erhalten. Nach diesem Wert sorgt ein starker Anstieg des E/A-Anteils, bewirkt durch eine längere E/A-CPU-Vorlaufzeit, für eine negative Grenzverbesserung oder anders ausgedrückt, die Sättigungsgrenze wurde überschritten. Obwohl ein MPL-Wert von 40 noch zu einer kurzen Wartezeit der Transaktion in der BOT-Warteschlange führt, ist die Antwortzeit für diesen Wert insgesamt besser als ohne eine Verzögerung.

Erklären läßt sich dieses Verhalten durch die hohe Belastung der CPU, die sich vor allem beim Transaktionstyp zwei auswirkt, der pro TA weit mehr CPU-Anforderungen als Typ eins stellt. Der Anstieg des FIX-Anteils geht auch hier auf die CPU-Wartezeit zurück.

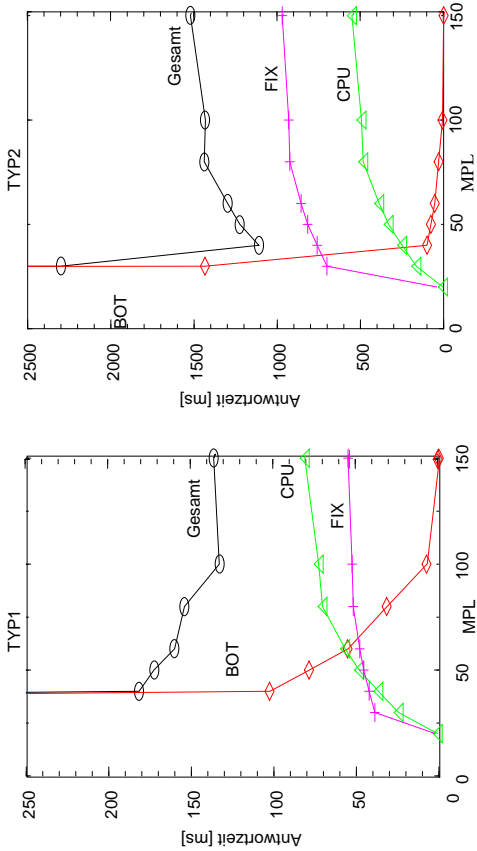


Abb. 5.12: Antwortzeitaufgliederung SMPL

Als Ergebnis der statischen MPL-Simulationen kann festgehalten werden, daß der optimale MPL-Wert für Typ 1 bei ca. 100 und für Typ 1 bei 40 liegt. Als ein Kompromißwert kann gemäß Abb. 5.12 der kleinere dieser beiden Werte angesetzt werden. Mit diesen Einstellungen erhält man Antwortzeiten von 180 / 1100 ms und TaInTime-Anteile von 28 / 30 %. Es bleibt festzuhalten, daß dies lediglich ein **Kompromißwert zu Gunsten** des langen Transaktions-typs ist.

Abb. 5.13 dokumentiert die Ergebnisse, die für die Simulationen mit GMPL-Regelung und LMPL-Regelungsstrategie erzielt wurden. Die mit den statischen MPL-Werten erhaltene Antwortzeit- und TAInTime-Kurve wurden zum Vergleich in die Grafik aufgenommen. Auf der Abszisse wird der statische MPL-Wert (SMPL) bzw. der Startwert des MPL für die dynamischen Verfahren (GMPL, LMPL) abgetragen. Die lokalen MPL-Werte wurden bei diesem Versuch mit den gleichen Werten initialisiert. Da sich beim lokalen Verfahren der globale MPL-Wert aus der Summe der lokalen MPL-Werte ergibt, wurden die Ergebnisse entsprechend der

Summe der Initialisierungswerte abgetragen. So wurden die Ergebnisse für die lokalen MPL-Startwerte 75 auf den Wert des gesamten MPL-Wertes von 150 dargestellt.

Sowohl die GMPL- als auch die LMPL-Strategie sorgen beim Antwortzeit-Verlauf für ein relativ gleichmäßiges Ergebnis. Die Antwortzeit für den Typ 1 liegen bei ca. 0.2 s und für den Typ 2 bei 1,2 s. Während die Antwortzeit-Ergebnisse für den Typ 2 im Vergleich mit den statischen Ergebnissen in fast allen Fällen niedriger liegen, ist dies für den Typ 2 genau umgekehrt. Dies schlägt sich noch offensichtlicher bei den TaInTime-Anteilen nieder. Im statischen Fall liegen die TaInTime-Raten für MPL-Werte größer als 50 nahe beieinander (der Abstand beträgt weniger als 5%). Die dynamischen MPL-Regelungen sorgen jedoch für ein „Auseinanderdriften“ der TaInTime-Anteile. So sinkt beim GMPL-Verfahren die TaInTime-Anteile des ersten Transaktionstyps auf 22%, während die Anteile für Typ 2 auf einen Wert von 37% anwächst.

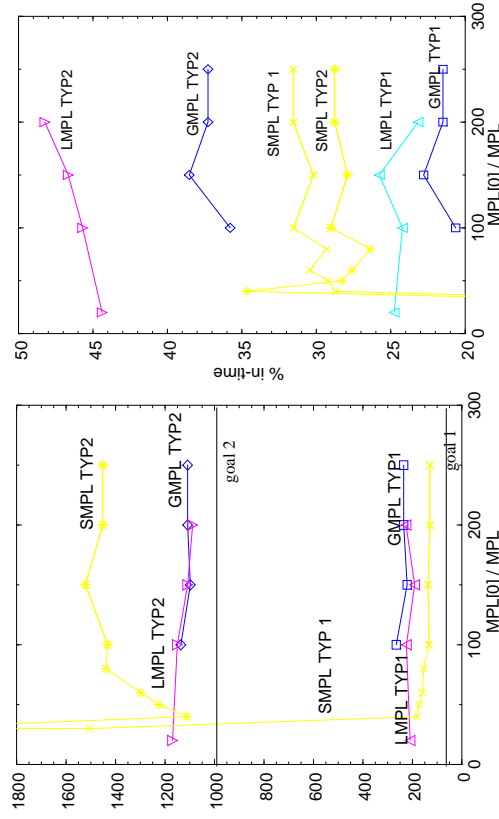


Abb. 5.13: Variation des MPL-Startwertes

Bemerkenswert ist auch das TaInTime-Verhalten des LMPL-Verfahrens. Die Reduktion der TaInTime-Anteile für den Typ 1 fällt geringer aus als bei der GMPL-Variante. Beim statischen Verfahren ergibt sich ein durchschnittlicher Wert von 30%. Dieser sinkt durch die Verwendung des LMPL-Verfahrens um 5-6%. Die Steigerung der TaInTime-Anteile für Typ 2 liegt jedoch bei ca. 20%.

In Abb. 5.14 werden die über dem gesamten Simulationszeitraum gemittelten MPL-Werte dargestellt. Diese Werte sind nur mit Einschränkungen zu bewerten, da es sich um Mittelwertwerte handelt. Die Werte geben nur Hinweise auf Stabilität des Verfahrens und schaffen eine Verbindung zu den Ergebnissen mit statischer MPL-Einstellung.

Obwohl die Resultate beim GMPL-Verfahren für verschiedene MPL-Startwerte dicht bei einander liegen, läßt sich als Optimum der Startwert von 150 bestimmen. Der durchschnittliche MPL-Wert lag bei dieser Simulation bei 36. Beim LMPL-Verfahren wurden ähnliche Ergebnisse für die Summe der lokalen MPL-Werte ermittelt. Die Summe variierte hierbei zwischen 35 und 38. Die Zusammensetzung des globalen MPL-Wertes bevorzugte in allen Fällen den Transaktionstyp 2. Die durchschnittlichen lokalen MPL-Werte für den ersten Typ lagen bei 14 und für den Typ 2 bei 21. Die CPU wurde somit auf MPL-Ebene in einem Verhältnis von 2:3 zwischen den beiden Typen aufgeteilt. Dieses Verhältnis steht im Widerspruch zur eigentlichen CPU-Lastsituation, die zu 66% aus Transaktionstyp 1 und nur 34% aus Transaktionstyp 2 besteht.

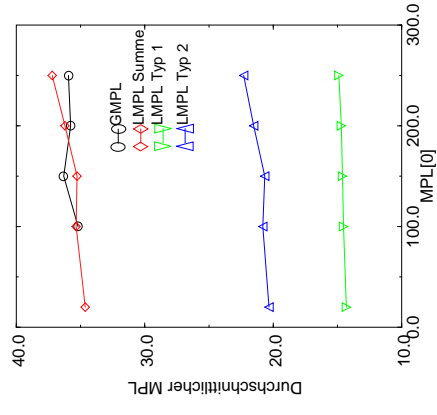


Abb. 5.14: Durchschnittlicher MPL, CPU-Engpaß 2 Typen

Da die Reduktion des MPL-Wertes nur bei einer Engpaßerkennung auftreten kann, muß folglich für den Typ 1 häufiger ein Engpaß erkannt werden als für den zweiten Typ. Diese These läßt sich auch mittels der Analyse der Antwortzeitanteile der beiden Typen weiter erhärten. Der Transaktionstyp 1 führt lediglich 5 Referenzen aus. Wie der Abb. 5.12 zu entnehmen ist

liegt der Anteil der E/A-Zeit bei ca. 40-50 ms, was ungefähr 25 bis 30% der gesamten Antwortzeit ausmacht. Der CPU-Anteil liegt für den relevanten MPL-Bereich von 40 bis 60 zwischen 20 und 55 ms (10 - 30%). Durch den geringeren E/A-Anteil und das Fehlen von weiteren Antwortzeit-Bestandteilen, wie der Sperrbearbeitung, ist die 25% Quote für die CPU-Engpaßerkennung häufig überschritten. Der zweite Typ leidet nicht unter diesem Problem, da der E/A-Anteil schon für niedrige MPL-Wert sehr hoch ist. Für den relevanten Bereich liegt die E/A-Zeit bei 750 bis 800 ms. Dies entspricht ca. 70 % der gesamten Antwortzeit.

Als Ergebnis der ersten Studie mit zwei Transaktionstypen kann somit festgehalten werden, daß das Verfahren mit lokalen MPL-Verfahren nicht invariant bzgl. der Struktur der auftretenden Transaktionstypen ist. Auch wenn die simulierte Situation durch das Fehlen von jeglichen Sperrkonflikten als rein synthetisch erscheinen mag, ist diese Beobachtung, die gegen die Verwendung von globalen Antwortzeitbestandteilen spricht, von Relevanz. Geht man davon aus, daß die hier beobachteten Strukturunterschiede in der Antwortzeitzusammensetzung bei einer größeren Anzahl von Transaktionstypen noch zahlreicher werden, so ist die Vorgabe von globalen für alle Transaktionen gültigen Antwortzeitanteils-Prozentsätzen als kritisch zu bewerten. Die Verwendung von globalen Antwortzeitanteils-Prozentsätzen ohne Berücksichtigung der vorhandenen Strukturunterschiede führt dazu, daß der oder die Transaktionstypen, die am meisten unter dem CPU-Engpaß „leiden“, durch Reduktion ihres lokalen MPL-Wertes „bestraft“ werden. Der eigentliche CPU-Engpaß-Versacher kann von der Regelung vollständig unberücksichtigt bleiben.

Die Ergebnisse des GMPL-Verfahrens sind anders zu interpretieren. Die Antwortzeiten liegen im Toleranzbereich. Die TaInTime-Raten sind für den zweiten Typ generell schlechter. Durch die Festlegung eines relativ geringen MPL-Wertes wird der erste Typ benachteiligt. Durch die lange Verweildauer des langen Transaktionstyps im System überschreiten viele der kurzen und häufig ankommenden Transaktionen ihre Antwortzeitvorgaben. Eine Wartezeit in der BOT-WS wirkt sich beim kurzen Transaktionstyp wesentlich stärker aus, da das Antwortzeitziel viel geringer ist als beim langen Typ. Beim LMPL-Verfahren konkurrieren die Transaktionen nicht um die MPL-Slots. Durch die Zuweisung von transaktionsklassenspezifischen MPL-Werten wird die TaInTime-Rate des ersten Typs leicht verbessert.

Dieses Verhalten ist insbesondere im Vergleich zum statischen Fall sehr auffällig. Es stellt sich die Frage, ob dieses Ungleichgewicht nicht durch andere Maßnahmen korrigiert werden kann. Diese Überlegungen werden im nächsten Abschnitt behandelt.

5.5 MPL-Regelung und Prioritätsalgorithmen

Durch die Festlegung des MPL-Werts wird lediglich festgelegt, wie viele Transaktionen insgesamt im System zur gleichen Zeit vorhanden sein dürfen. Durch die Verwendung von lokalen MPL-Werten kann eine Last-Mix-Kontrolle eingeführt werden. Wie im letzten Abschnitt erläutert, ist jedoch auch dieses Verfahren mit Problemen behaftet. Deshalb wurden Untersuchungen dahingehend unternommen, ob das statische und das globale MPL-Verfahren nicht durch Prioritäten noch optimiert werden können. Unter Optimierung wird an dieser Stelle eine zielabhängige Gleichbehandlung der Transaktionsklassen verstanden. In den Simulationen soll somit eruiert werden, ob die beobachtete Ungleichbehandlung der strukturverschiedenen Typen aus dem letzten Abschnitte überwunden werden kann. Als ersten Schritt in diese Richtung wird eine Simulationsreihe mit den zwei aussichtsreichsten Prioritätsmethoden durchgeführt. Als zweiter Schritt stellt sich die Frage, ob eine prioritätsorientierte Steuerung nur einzeln für die BOT-WS und die CPU-Request-Warteschlange oder gemeinsam verwendet werden soll.

Im folgenden werden nur die Prioritätsvorgaben mit der Earliest-Deadline-Methode (ED) und der Verwendung von Performance-Indexes (PI) verwendet (vgl. [FNG93]). Bei beiden Prioritätsalgorithmen werden die Antwortziele berücksichtigt. Dies geschieht beim ED-Verfahren durch die individuelle Berechnung der Deadline einer einzelnen Transaktion aus der Ankunfts- bzw. Startzeit und der vorgegebenen Antwortzeit für die jeweilige Transaktionsklasse. Die Priorität einer Transaktion wird im Verlaufe der Zeitablaufes immer größer. Beim PI-Verfahren werden die Prioritäten immer für eine ganze Transaktionsklasse bestimmt. Die Berechnung erfolgt unter Berücksichtigung der vorgegebenen Antwortzeiten und der Lastsituation bzw. der Antwortzeit-Ergebnisse von beendeten Transaktionen der jeweiligen Klasse. Die Prioritäten der Transaktionsklassen bleiben beim PI-Verfahren bis zu ihrer Neuberechnung konstant.

Für die Simulation wurden die gleichen Parameter wie im letzten Abschnitt gewählt. Die Prioritätssteuerung gilt für beide Fälle sowohl für die BOT- als auch für die CPU-Warteschlange.

5.5.1 Ergebnisse bei statischer MPL-Einstellung

Die Abb. 5.15 zeigt die mit dem SMPL-Verfahren erhaltenen Ergebnisse. Die besten Antwortzeit-Ergebnisse liefert die ED-Methode. An zweiter Stelle liegt das Verfahren mit First-Come-First-Serve (FCFS) Verarbeitung und an letzter Stelle das PI-Verfahren.

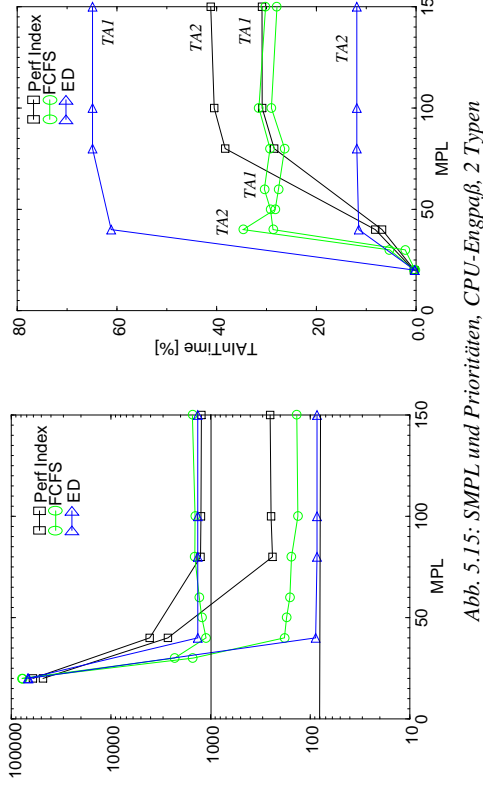


Abb. 5.15: SMPL und Prioritäten, CPU-Engpaß, 2 Typen

Die Unterschiede fallen für beide Typen unterschiedlich stark aus. Die TaktTime-Kurve deckt weitere Unterschiede auf. Durch das ED-Scheduling wird in diesem Fall eindeutig der Typ 1 bevorzugt. Die kurze Transaktion erhält aufgrund der kürzeren Antwortzeit-Vorgaben eine schneller wachsende Priorität. Dies steht im Einklang mit Resultaten aus anderen Studien (vgl. [AG89], [HLC91]). Die Ergebnisse der Performance-Index-Regelung sind differenziert zu bewerten. Für den Typ 2 unterscheiden sie sich bei den verschiedenen Varianten kaum. Die besten TaktTime-Anteile werden durch die FCFS-Variante ab einem MPL-Wert von 50 erzielt. Für die Regelung mit PI werden akzeptable Ergebnisse erst ab einem MPL-Wert von 75 erreicht. Für den Transaktionstyp 1 schneidet die PI-Regelung in Bezug auf die Antwortzeit als schlechteste Strategie ab. Die TaktTime-Anteile liegen hier jedoch ab einem Wert von 75 höher als bei der FCFS-Variante.

Dieses Verhalten ist nur zu erklären, wenn man sich das Funktionsprinzip der Prioritätensteuerung mit dem Performance-Index vergegenwärtigt. Mit dieser Regelung bekommt die Transaktionsklasse die höhere Priorität, die die Zielvorgabe um einen relativen Wert **mehr** verfehlt.

Die Folge ist, daß im nächsten Intervall die Transaktionen mit der momentan höchsten Priorität bevorzugt gestartet und mit CPU-Zeit bedient wird. Unterstellt man, daß alle Transaktionen in dem selben Intervall beendet werden, so schwankt das System zwischen 100% TaInTime-Anteilen für Typ 1 in diesem Intervall und 100% TaInTime-Anteilen für Typ 2 im nächsten Intervall. Dies tritt ein, da nach Beendigung des betrachteten Intervalls der andere Transaktionstyp einen höheren PI erhält. Diese Situation entspannt sich beim vorliegenden Fall etwas, da die Antwortzeit des Typ 2 über 1 s liegt und somit einige Transaktionen in zwei Arbeitsintervalle verarbeitet werden. Das eben beschriebene Verhalten ist jedoch auch hier erkennbar. Die Regelung mit PI sorgt somit dafür, daß für hochprioritäre TA die Ressourcen reserviert sind. Der nicht benötigte Teil der CPU-Kapazität kann nun dann genutzt werden, wenn noch weitere Bearbeitungslots zur Verfügung stehen. Aus diesem Grund steigt die TaInTime-Anteile auch erst für höhere MPL-Werte auf einen akzeptablen Wert an. Darüberhinaus ist noch anzumerken, daß die klassenorientierte Prioritätssteuerung bei Verwendung von Prioritäten auf CPU-Ebene auch noch Probleme aufwirft. Durch die Umstellung in den einzelnen Berechnungspunkten, werden ggfs. im Verarbeitungsprozeß (nach BOT) befindliche **lange TA** verzögert. Die Prioritäten sollten somit in künftigen Implementationen den Transaktionen beim Beginn der Transaktion individuell zugewiesen werden.

5.5.2 Ergebnisse beim GMPL-Verfahren

Die Resultate, die unter Verwendung der Prioritätsalgorithmen und der GMPL-Strategie erzielt wurden, zeigt die Abb. 5.16. Die Resultate sind denen der statischen Simulationen ähnlich. Die besten Antwortzeiten für den Transaktionstyp 1 werden durch die ED-Variante erzielt. Sie liegen in etwa gleicher Höhe wie beim SMPL-Verfahren mit FCFS. Die Resultate für den langen Transaktionstyp entsprechen, mit Ausnahme der PI-Variante, den mit dem SMPL-Verfahren erzielten Antwortzeiten. Beim SMPL-Verfahren mit FCFS liegen die TaInTime-Anteile für beide Typen eng bei einander. Es werden Werte von ca. 30% erreicht. Als genereller Trend beim GMPL-Verfahren ist festzustellen, daß die Gleichbehandlung der Klassen nicht mehr gegeben ist. Zwischen den TaInTime-Anteilen für Typ 1 und 2 liegen in allen Fällen, wiederum mit Ausnahme des PI, ca. 15%. Dies läßt sich auf einen geringen durchschnittlichen MPL-Wert zurückführen.

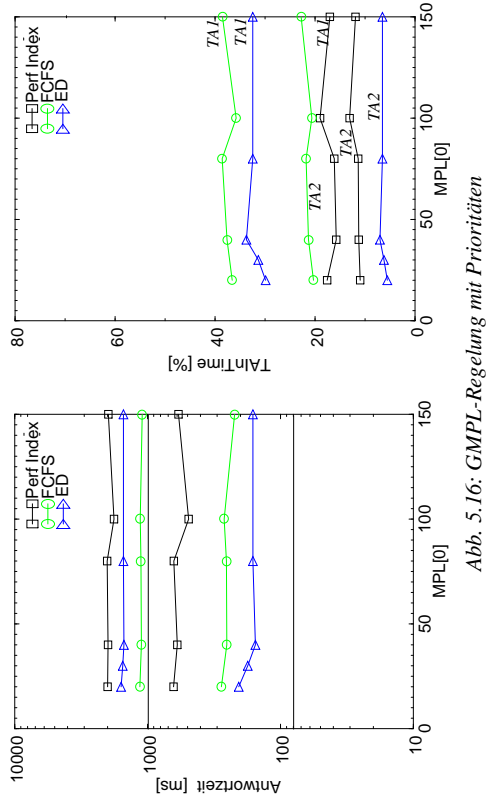


Abb. 5.16: GMPL-Regelung mit Prioritäten

Die Regelung erzielt einen durchschnittlichen MPL-Wert von 33 bei Verwendung der ED-Strategie und 43 bei Verwendung der PI-Strategie. Die ED- und FCFS-Strategie bevorzugt auch hier die kurze Transaktionen. Die PI-Strategie schneidet hier sowohl bzgl. Antwortzeit als auch bzgl. TaInTime-Anteile als schlechteste ab. Gegenüber der Variante mit FCFS-Steuerung wird hier jedoch eine Nivellierung der beiden Typen bzgl. TaInTime-Anteilen erreicht. Dies ist jedoch unter Berücksichtigung einer mittleren Antwortzeit für Typ 1 von ca. 0.7 s kaum als wirkliche Alternative anzusehen.

5.5.3 Ergebnisse bei unterschiedlicher Verwendung von Prioritäten

Die bisher betrachteten Varianten benutzten die Prioritäten sowohl auf BOT- als auch auf CPU-Ebene. In diesem Abschnitt wird untersucht, ob Prioritäten durchgängig für CPU und BOT-WS oder lediglich in einem der beiden Bereiche genutzt werden sollten. Die Untersuchungen wurden für das ED- und das PI-Verfahren vorgenommen. Ein Vermischen der beiden Strategien (d.h. kombinierter Einsatz in unterschiedlicher Bereichen) wurde nicht vorgenommen.

Verwendung des Earliest-Deadline-Verfahrens.

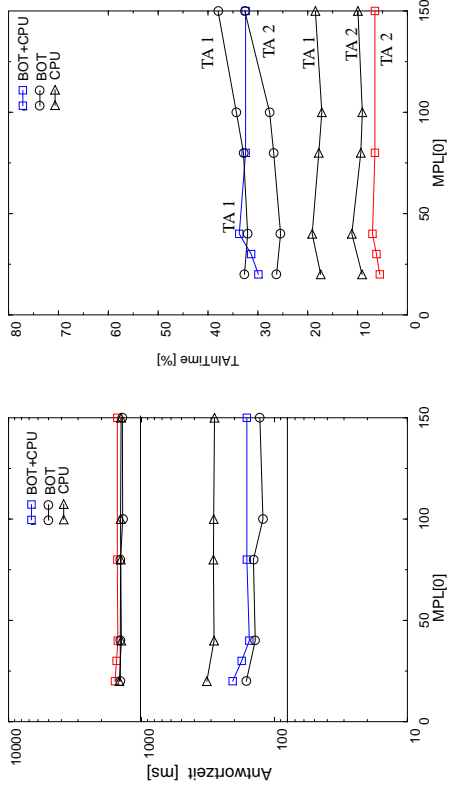


Abb. 5.17: ED-Verfahren in verschiedenen Warteschlangen
GMPL-Regelung, CPU-Engpaß

Abb. 5.17 zeigt die Ergebnisse der Simulationen für das ED-Verfahren. Die Antwortzeiten liegen für den Transaktionstyp 2 eng zusammen. Die logarithmische Skala täuscht jedoch darüber hinweg, daß zwischen den einzelnen Varianten durchaus Unterschiede von bis zu 200 ms auftreten. Das für alle MPL[0]-Werte beste Resultat wird bei ausschließlicher Verwendung auf BOT-Ebene erreicht. Für den Transaktionstyp 1 stellt sich die Situation analog dar. Die ausschließliche Verwendung auf BOT-Ebene liefert die kürzesten Antwortzeiten und die besten TaInTime-Anteile. Es lassen sich für die Verwendung der ED-Strategie weitere Beobachtungen festhalten. Die Ergebnisse lassen für die beiden Typen bei Verwendung von Prioritäten allein auf CPU-Ebene und kombiniert mit der BOT-WS kein eindeutiges Urteil zu. Beim kurzen und CPU-intensiven Transaktionstyp wirkt sich die Verwendung von FCFS für die BOT-WS und ED für die CPU-WS sehr negativ aus. Dies liegt weniger an möglichen Überholungen von Transaktionen auf CPU-Ebene, sondern an der „Anfälligkeit“ des kurzen Typs für BOT-Wartezeiten. Diese werden bei den anderen Verfahren vermieden. Beim langen Transaktionstyp wirken sich BOT-WS-Zeiten nicht so stark aus.

Verwendung des Performance-Index-Verfahrens.

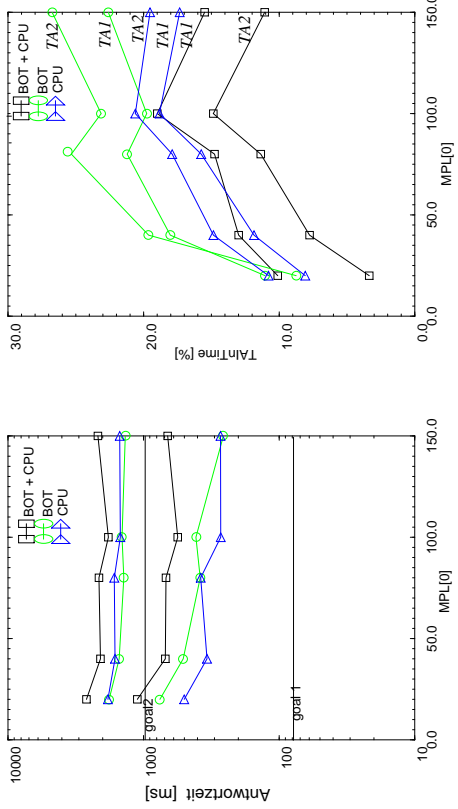


Abb. 5.18: PI-Verwendung in verschiedenen Warteschlangen
GMPL-Regelung, CPU-Engpaß

Das beste Ergebnis liefert die reine Verwendung auf BOT-WS-Ebene. Die ausschließliche Verwendung auf CPU-Ebene liefert zwar für den Typ 1 eine durchschnittlich bessere Antwortzeit, ist aber bzgl. der TaInTime-Anteile generell schlechter. Bemerkenswert ist die Abhängigkeit der Ergebnisse vom Anfangswert. Die Ergebnisse ohne Prioritätsteuerung sind wesentlich stabiler. Dies fällt vor allem im Hinblick auf die TaInTime-Anteile auf. Dies läßt sich auch auf das stark alternierende Verarbeitungsverhalten, wie es im vorletzten Paragraphen erläutert wurde, zurückführen.

Die zielorientierte Optimierung des Verarbeitungsmixes konnte mit der einfachen Einbindung des PI nicht erreicht werden. Es bietet sich aber eine verfeinerte Vorgehensweise an. Die Engpaßressource MPL-Slot kann mittels der Prioritäten **kontingentierte** werden. Dies bedeutet, daß die Transaktionsklasse mit dem höchsten Priorität auch die meisten Slots erhält. Es wird aber verhindert, daß die Klassen mit der höchsten Priorität uneingeschränkt mit MPL-Slots versorgt wird. Bei einer Kontingentierung geschieht diese Vorzugsbehandlung nur bis das zugeteilte Kontingent verbraucht ist. Kommen weitere Transaktionen der Klasse an, so müssen sie sich entsprechend einer zur wählenden (z.B. FCFS) Strategie um noch freie allgemeine (dies sind insbesondere die nicht von anderen Klassen benötigten MPL-Slots) Slots bemühen.

6. Lastkontrolle bei Sperrengpässen

Die Erkennung eines CPU-Engpasses stellt im Vergleich mit der Bestimmung eines Sperrengpasses eine klar umrissene Aufgabe dar. Durch das CPU-Auslastungsmaß hat man einen feste Maßgröße zur Orientierung. Die Antwortzeit-Methode läßt sich durch die Einbeziehung dieser Größe gut ergänzen. Die Behandlung von Sperrengpässen stellt demgegenüber ein höheres Maß an Komplexität dar. Es existieren für die Softwarekomponente des Sperrens keine festen Maße für die Bestimmung eines möglichen Engpasses. Wie bereits im Abschnitt 3erörtert konkurrieren die Transaktionen beim Sperrverfahren um den Objektzugriff. Während reine Lesezugriffe miteinander kompatibel sind, erfordern Schreibzugriffe einen exklusiven Zugriff. Es lassen sich zwei wichtige Faktoren für die Verarbeitung einer Transaktion festhalten :

- die Häufigkeit der Konkurrenz
- die Länge der Wartezeiten auf Sperren

Diese beiden Faktoren bestimmen im Bereich der Sperrengpaßverwaltung die Verweildauer und somit auch den Durchsatz von Transaktionen im System. In den Simulationsstudien werden drei verschiedene Methoden bzw. Maßgrößen verwandt, um einen Engpaß zu erkennen. Die Regelung erfolgt nach folgenden Gesichtspunkten (vgl. auch Abschnitt 3) :

- (1) der Bestandteil der Antwortzeit, die für die Sperrbearbeitung notwendig war, sollte einen bestimmten Anteil nicht überschreiten
- (2) die Anzahl der Konflikte an der Gesamtheit aller Referenzen (CNFL-Rate) sollte einen bestimmten Wert nicht überschreiten.
- (3) der Quotient der Anzahl der Seiten, die von aktiven (nicht wartenden) Transaktionen gehalten werden, und der Summe aller gehaltenen Seiten (WKM-Rate) sollte einen bestimmten Wert nicht überschreiten.

Bereits im Vorfeld der Untersuchungen zeigte sich, daß die ursprüngliche Implementierung modifiziert werden mußte. Bei der Behandlung des CPU-Engpasses wurde nach dem Erkennen eines BOT-WS-Engpasses erst getestet, ob nicht bereits eine bestimmte CPU-Auslastung überschritten wurde. In der ursprünglichen Implementation für den Regelkreis BOT-Sperrengpaß wurde ein ähnliches Verfahren verwandt. Es wurde aufgrund des prozentualen Anteils der Antwortzeit des letzten Intervalls geprüft, ob diese z.B. 90 % der für die Sperranforderung vorgegebenen Antwortzeit von 25 % der gesamten Bearbeitungszeit überstieg. Falls dies der

Fall war, wurde eine Erhöhung des MPL-Wertes nicht gestattet. Die prozentuale Erkennung funktioniert im Normallast und Hochlastfall. Tritt jedoch eine Überlast auf, so vergrößert sich durch den Verarbeitungsrückstau die Warteschlange. Entsprechend erhöht sich auch der BOT-WS-Anteil der Antwortzeit. Ist der BOT-WS-Anteil hinreichend groß und folglich auch die gesamte Summe der Antwortzeiten, so ist der prozentuale Anteil der Sperrbearbeitung an der gesamten Antwortzeit sehr gering. Die Methode garantiert also nicht, daß die MPL-Erhöhung abgewiesen wird. Der Effekt ist die komplette *Verstopfung* des Systems. Durch den zu hoch eingestellten MPL-Wert erhöhen sich die Konflikte und somit auch die Verweildauer der Transaktionen im System. Dies bewirkt eine längere Wartezeit in der BOT-WS, was wiederum zur BOT-WS-Engpaßerkennung und wieder zur MPL-Erhöhung führt. Deshalb trat an dieser Stelle der einfachen Antwortzeiteanteil-Analyse eine Kombination mit der CNFL-Rate. Die MPL-Erhöhung wird somit nur dann gestattet, wenn der Antwortzeiteanteil **und** die CNFL-Rate unter einem bestimmten Schwellwert bleiben.

Die Simulationen um den CPU-Engpaß haben folgendes Vorgehen für die weiteren Untersuchungen nahegelegt. In der ersten Phase wird mittels der Variation des Schreibanteils bestimmt, für welche Werte die Last eine Normallast, Hochlast oder Überlast darstellen. Das Verhalten wird hierbei mit den dynamischen Verfahren analysiert. Der Normallastfall ist für die Regelungskomponente als unkritisch zu bewerten. Es werden auch mit schlechten Parametern ähnlich gute Ergebnisse wie im besten statischen Fall erreicht. Zur Untersuchung werden aus den sogenannten *Eichungsläufen* zwei Szenarien ausgewählt. Diese sollen den anderen beiden Fällen entsprechen. Der Hochlastfall und der Überlastfall unterscheiden sich dahingehend, daß in letzterem das System nicht mehr in der Lage ist, die Last vollständig zu verarbeiten. Im Hochlastfall operiert das System nahe an der maximalen Verarbeitungs- bzw. Kapazitätsgrenze. Dieses Vorgehen hat den Hintergrund, daß der Hochlastfall weitere Aufschlüsse bezüglich des Vergleiches der erzielbaren Ergebnisse der dynamischen Strategien liefern soll. Der Überlastfall simuliert zu erwartende Lastspitzen und soll unter anderem die Frage beantworten, ob die neue Überlastsicherung zur Verhinderung des *Davonlaufens* des MPL-Wertes die gewünschten Ergebnisse liefert.

Die Zielantwortzeiten wurden für die einzelnen Transaktionen mit Absicht sehr knapp am rechnerischen Minimum angesetzt. So wird gesichert, daß die Regelungskomponente sehr oft in die Analysephase eintritt. Auf diese Weise wird gewährleistet, daß der MPL-Wert trotz

einer sehr häufigen Regelung innerhalb eines optimalen Einstellungskorridors für den MPL-Wert bleibt.

Die beiden Szenarien werden dann folgenden Untersuchungen unterzogen. Es wird zuerst der beste statische MPL-Wert für die Szenarien ermittelt. Danach werden für die dynamischen Verfahren die besten Parameter (Antwortzeitanteil, CNFL-Rate, WKM-Rate) für beide Szenarien ermittelt. Die so gewonnenen Parameter werden im zweiten Schritt mit den Ergebnissen aus einem Lauf mit 2 Transaktionstypen und stabilem Ankunftsverhalten verglichen.

In den beiden letzten Abschnitten werden die Ergebnisse einer Simulationsreihe mit 2 Transaktionstypen und Lastspitzen und einer Simulationsreihe mit einer realen Datenbanklast dargestellt.

6.1 Simulationen mit einem Transaktionstyp

Die Reihe der Simulationen wird mit der Betrachtung von einzelnen Transaktionstypen begonnen. Es werden zwei verschiedene Transaktionstypen untersucht. In den einzelnen Simulationen wird jedoch immer nur ein Transaktionstyp verarbeitet. Die beiden Transaktionstypen unterscheiden sich in der Anzahl der Referenzen. Die Anzahl der Referenzen einer Transaktion bleiben im Simulationszeitraum konstant. Die kurze Transaktion referenziert nur 5 Objekte. Die lange Transaktion generiert 50 Referenzen.

Die Simulationszeit betrug 1000 s. Bei den gewählten Ankunftsraten wurden somit in den Simulationen mit dem kurzen Transaktionstyp ca. 100.000 Transaktionen mit insgesamt 500.000 Referenzen generiert. Die Werte für die Simulationen mit dem langen Transaktionstyp liegen bei ca. 20.000 TA mit 1.000.000 Referenzen.

Für die kurze TA war eine Anpassung der Datenbankpartition notwendig. Da die Transaktionen nur sehr kurz im System verweilen, werden bei einer großen Anzahl von Seiten (z.B. 10.000) eine sehr große Anzahl von Transaktionen notwendig (> 2000 tps), um einen Sperrengpaß herbeizuführen. Aus diesem Grund wurde die referenzierte Partition auf 1000 Objekte verkleinert. Dies entspricht ein Seitenanzahl von 100. Um das Verhältnis Seitenanzahl der Partition zu Seitenpuffer in realistischen Dimensionen zu halten, wurde die Puffergröße auf 20

Seiten gesenkt. Dies entspricht wieder dem bisher verwendeten Verhältnis von 1:5. Die lange Transaktion operiert auf einer Datenbankpartition mit 100.000 Objekten und 10.000 Seiten. Der Puffer wurde auf 2000 Seiten eingestellt. Die Parameter werden in Tabelle 6.1 zusammengefaßt.

Relation	Parameter	Simulationsreihe 1	Simulationsreihe 2
Anzahl Objekte		1.000	100.000
Blockungsfaktor		10	10
Transaktionen			
Anzahl Typen		1	1
Zielantwortzeiten (s)		0.1	1.0
Ankunftsrate		100	20
Referenzen		5	50
Schreibanteil		variiert	variiert
Zugriffsverteilung		1 Partition	1 Partition
Spergranulat		Seiten	Seiten
Spermethode		2-phasesiges Sperren	2-phasesiges Sperren
Puffergröße (Seiten)		20	2000
Logging		JA	JA

Tabelle 6.1: Grundparameter der Sperr-Engpaß-Simulationen mit einem Typ

6.1.1 Bestimmung der Szenarien

Die initialen Parameter wurden für die verschiedenen Verfahren entsprechend Tabelle 6.2 gewählt. Die Werte für den Antwortzeit-Anteil wurden aus den bereits in anderen Simulations-

Indikator	Antwortzeitanteil	CNFL-Rate	WKM-Rate
Wert	25 %	10 %	1.3

Tabelle 6.2: Initialwert für Sperrengpaßerkennung

läufen durchgeführten Simulationen entnommen [Kr94]. Die CNFL-Rate wurde unter Berücksichtigung der Erkenntnisse von im Vorfeld dieser Untersuchung vorgenommenen Experimenten gewählt. Die WKM-Rate wurde mit dem in der Literatur als optimalen Wert deklarierten von 1.3 belegt [MW92].

In den Abb. 6.1 und Abb. 6.3 werden die Ergebnisse für die Variation des Schreibanteils der Transaktionen gezeigt. Der CNFL-Raten-Wert gilt auch für die MPL-Erhöhungsbemühungen bei der antwortzeitorientierten Erkennung. Der Korridor beim Test wurde relativ knapp auf 5 % des Steuerungswertes gelegt. Dies bedeutet für die antwortzeitbezogene BOT-WS-Engpaßregelung, daß eine MPL-Erhöhung nur stattfindet, falls der Sperrantwortzeitanteil geringer als 23 % und die CNFL-Rate kleiner als 9,5 % ist.

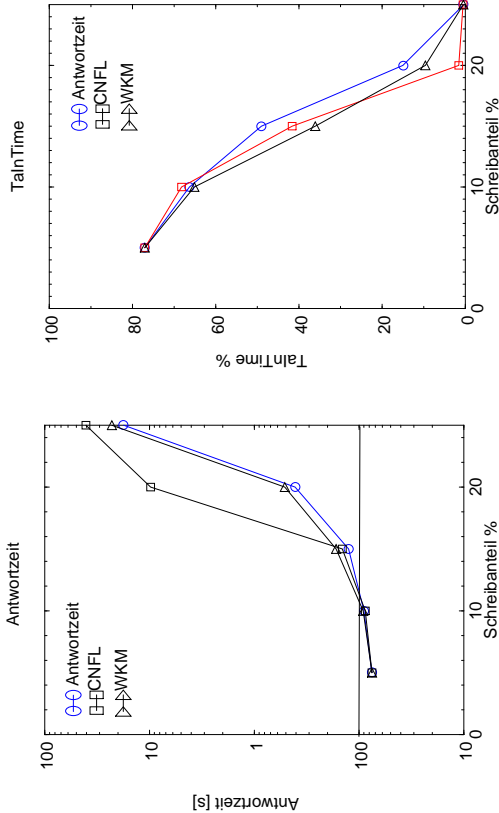


Abb. 6.1: Variation des Schreibanteils, Typ 1, Sperrengpaß 1 Typ

Der Schreibanteil wurde für den kurzen Transaktionstyp von 5 % bis 25 % variiert. Die gewonnenen Ergebnisse zeigt die Abb. 6.1. Das System zeigt bis zu einer Änderungsrate von 15 % ein stabiles Verhalten. Ab einem Wert von 20 % überschreitet die Antwortzeit die durchschnittliche Zielvorgabe um ein Vielfaches. Es wurden aus diesen Szenarien das Szenario mit 10 % und 25 % für weitere Untersuchungen ausgewählt. An Stelle des 25 %-Szenarios hätte auch das 20 %-Szenario als Überlastsituation gewählt werden können. Dies erschien aber nicht ratsam, da nicht von vorneherein davon ausgegangen werden kann, daß nicht durch eine optimale statische MPL-Wahl das Ergebnis wesentlich verbessert werden kann.

Im 10 %-Szenario liegen die Antwortzeiten eng beisammen. Sie liegen alle bei ca. 0.09 s. Mit dieser Antwortzeit werden TaInTime-Anteile von 70 % erreicht. Im 25 %-Szenario sinken die TaInTime-Anteile auf fast 0 % ab. Die Antwortzeit liegen für die drei Varianten zwischen 22 s und 41 s und überschreiten die Zielantwortzeit von 0.1 s somit um ein Vielfaches.

In der Abb. 6.2 werden weitere Informationen über die Simulationläufe gezeigt. Die erste der beiden Grafiken zeigt die durchschnittlichen MPL-Werte. Diese nehmen bei zunehmendem Schreibanteil immer weiter ab. Die andere Grafik dokumentiert das Durchsatzverhalten. Hier

wird gezeigt, wie viele der ankommenden Transaktionen (insgesamt 40.000) auch tatsächlich am Ende der Simulation das System wieder verlassen haben. So zeigt sich, daß für die CNFL-Raten bezogene Erkennung bereits ein Durchsatzproblem bei einem 20 % Schreibanteil vor- kommt, während bei den anderen beiden Erkennungsarten dies erst ab 25 % der Fall ist.

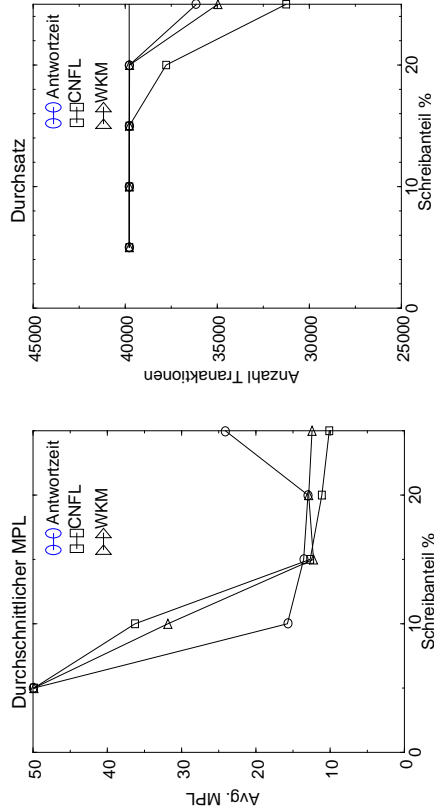


Abb. 6.2: Durchschnittlicher MPL-Wert und Durchsatz Typ 1, Sperrengpaß 1 Typ

Die durchschnittlichen MPL-Werte verlaufen erwartungsgemäß umgekehrt proportional zum Schreibanteil. Lediglich beim antwortzeitorientierten Verfahren ist für die Änderungsrate von 25 % ein abweichendes Verhalten festzustellen. Aufgrund eines sehr hohen „Rücktausch“ in der BOT-WS, ist das antwortzeitorientierte Verfahren nicht in der Lage, den eigentlichen Sperrengpaß zu erkennen.

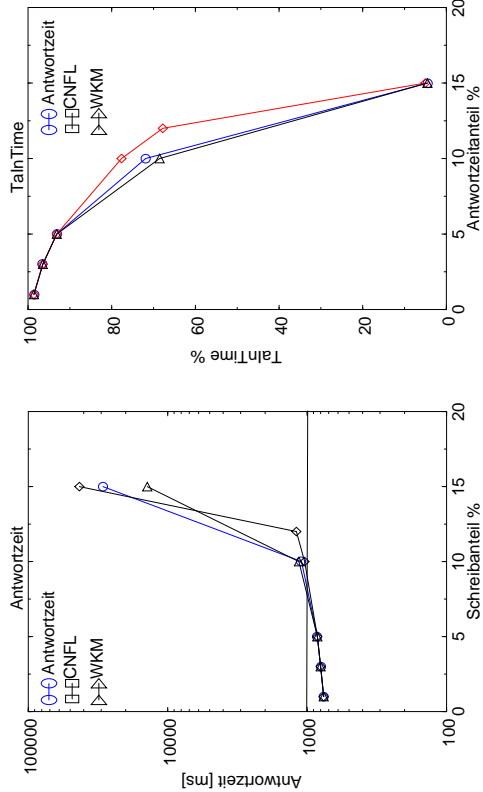


Abb. 6.3: Variation des Schreibanteils, Typ 2, Sperrengpaß 1 Typ

Für den langen Transaktionstyp wurde der Schreibanteil von 2 % bis 15 % variiert. Die Ergebnisse stellt die Abb. 6.3 dar. Für die weiteren Untersuchungen wurden die beiden Szenarien mit 10 % und 15 % Prozent ausgewählt. Im 10 % Schreibanteil Fall erreichen noch bis zu 80 % der Transaktionen die Zielantwortzeit von 1 s. Diese Quote sinkt beim 15 % Fall in der Promille Bereich ab. In Abb. 6.4 werden die durchschnittlichen MPL-Werte und der Durchsatz dokumentiert. Auch hier gilt für den Überlastfall, daß viele Transaktionen am Ende der Simulation noch nicht abgearbeitet sind.

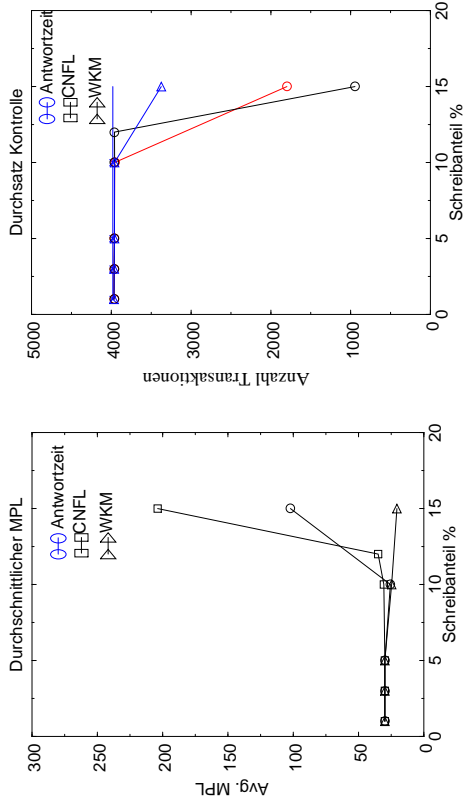


Abb. 6.4: Durchschnittlicher MPL und Durchsatz Typ 2 - Sperrengpaß 1 Typ

In den folgenden Experimenten werden die Durchsatzanalysen nicht mehr explizit für jeden Lauf dokumentiert. In den Simulationen konnte festgestellt werden, daß in den Überlastfällen eine minimale durchschnittliche Antwortzeit auch ein Durchsatzmaximum mit sich brachte.

Die MPL-Entwicklung deutet schon hier auf ein Problem hin. Es ist generell davon auszugehen, daß bei steigendem Schreibanteil auch die MPL-Werte sinken, um den Konfliktzuwachs auszugleichen. Dies ist ab einem Änderungsanteil von 10 % nicht mehr der Fall. Ursächlich ist dieses Verhalten durch zwei Möglichkeiten zu begründen. Die Sperrengpaßerkenntnis wird dadurch getäuscht, daß nicht mehr alle Transaktionen bearbeitet werden können. Es erfolgt somit keine bzw. nicht oft genug eine Reduzierung des MPL-Wertes. Als zweite denkbare Alternative ist die möglicherweise zu hohe Wahl der CNFL-Rate bei der BOT-WS-Engpaßbehandlung. Welche Alternative sich als zutreffend erweist, läßt sich mit den Ergebnissen aus den folgendem Abschnitt bestimmen.

6.1.2 Bestimmung des optimalen statischen MPL-Wertes

Es empfiehlt sich, für die beiden jeweils gewählten Szenarien die jeweils optimalen oder den optimalen statischen Wert zu bestimmen. Die reine Betrachtung der Antwortzeit und den TaInTime-Anteile wird hier noch um einige Teilbetrachtungen ergänzt. Diese sollen Aufschluß geben über die Werte für die drei möglichen Engpaßindikatoren. Die Antwortzeiten der Transaktionen werden in ihre Bestandteile aufgesplittet. Es lassen sich somit qualitative Aussagen über die Sperranteile und deren Entwicklung an der gesamten Antwortzeit machen. Diese Angaben sind vor allem für das antwortzeitbezogene Verfahren von Interesse. Sie geben Anhaltspunkte für die richtige Wahl der Prozentsätze und verifizieren ggfs. die Werte aus den Vorüberlegungen. Die Werte für die CNFL-Rate und die WKM-Rate sollen unter anderem erste Anhaltspunkte für die Bewertung dieser Größen als Lastbeschreibungsindikatoren liefern.

Die im letzten Abschnitt ausgewählten Fälle werden im folgenden Abschnitt auch in Kurzform genannt. Die Simulationen werden für den kurzen Typ mit der Nummer 1 und je nach 10 % oder 25 % Änderungsanteil mit a oder b betitelt. Analog wird der lange Typ mit Nummern 2a bzw. 2b versehen.

Bestimmung für den kurzen Transaktionstyp

Die Ergebnisse der Simulationen mit statischer MPL-Regelung für den kurzen Transaktionstyp werden in der Abb. 6.5 dargestellt. Es zeigt sich, daß der Fall mit 10 % Änderungsrate unkritisch bzgl. der MPL-Einstellung ist. Die Wahl des MPL-Wertes kann lediglich zu geringfügig gewählt werden. Die TaInTime-Anteile liegen hier im besten Fall bei 97 %. Dies liegt sehr nahe am Optimum.

Der Fall mit 25 % Änderungsanteil ist als kritisch bzgl. der Sperren zu nennen. Die Antwortzeitkurve findet ein Minimum bei einem MPL-Wert von 20. Die TaInTime-Anteile werden nur der Vollständigkeit halber aufgeführt. Die Quote liegt für den 25 % Fall so niedrig, daß sie von geringer Aussagekraft sind.

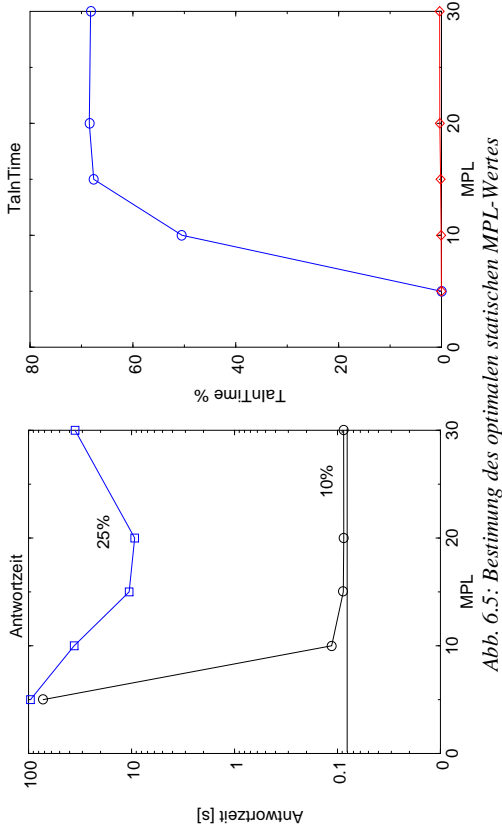


Abb. 6.5: Bestimmung des optimalen statischen MPL-Wertes

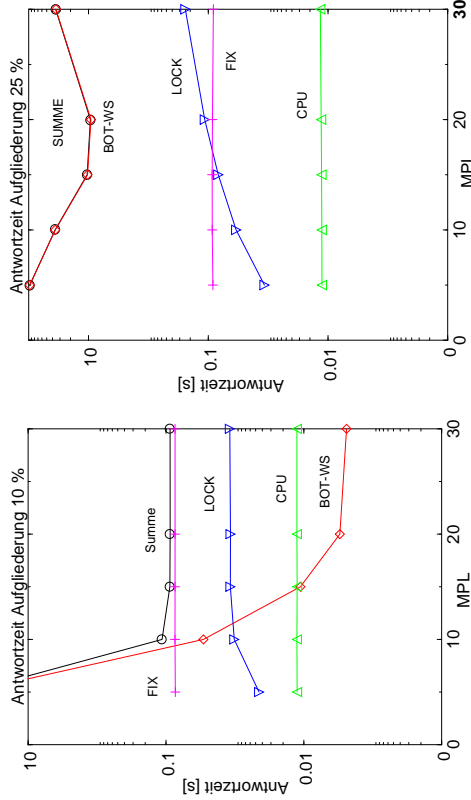


Abb. 6.6: Antwortzeit -Zusammensetzung SMPL, Typ 1, Sperrengpaß 1 Typ

In Abb. 6.6 werden die Antwortzeit-Zusammensetzung der beiden Simulationläufe dargestellt. Die Beobachtungen werden hier bestätigt. Der Sperranteil wird beim Fall mit 10 % Schreibanteil ab einem Wert von 10 nicht mehr erkennbar größer. Aufgrund der geringeren Anzahl der Referenzen bewegt er sich zudem auf einem sehr niedrigen Niveau. Bei einer Antwortzeit von ca. 87 ms liegt der Sperrbearbeitungsanteil mit ca. 11 ms bei 13 %. Beim zweiten Fall erkennt

man die Gründe für das Ansteigen der gesamten Antwortzeit jenseits des MPL-Wertes von 20. Der reine Sperrbetrag übersteigt die eigentliche Zielantwortzeit von 0.1 s. Diese Kurve (LOCK) flacht auch nicht ab, sondern steigt stetig an. Der Sperrbearbeitungsanteil liegt aufgrund des sehr hohen BOT-WS-Antwortzeitanteils lediglich bei 0.3 % der gesamten Antwortzeit.

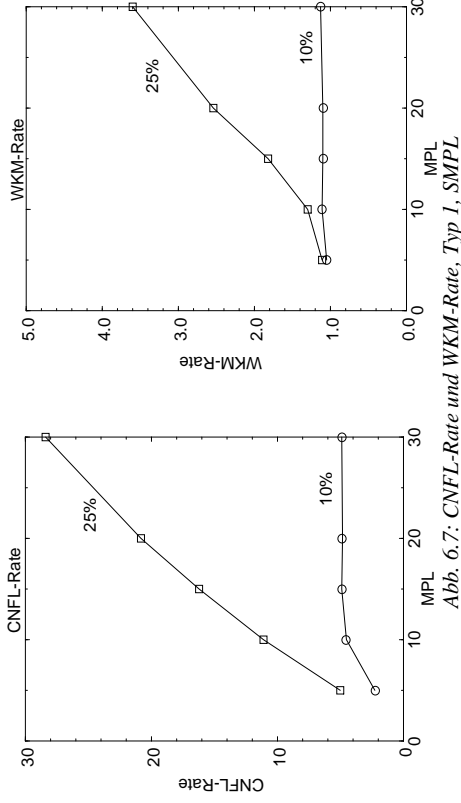


Abb. 6.7: CNFL-Rate und WKM-Rate und WKM-Rate, Typ 1, SMPL

Die beiden anderen Engpaßindikatoren entwickeln sich bei steigenden MPL-Wert gemäß Abb. 6.7. Es zeigt sich auch hier wieder, daß der 10 % Fall unkritisch bzgl. des Sperrbereiches ist. Die CNFL-Rate von 4 % und eine WKM-Rate von 1.1 sind als fast konstant anzusehen. Im Überlastfall zeigen sich dabei jedoch Unterschiede. Sowohl die Konflikt rate als auch die WKM-Rate sind streng monotone Graphen über dem MPL-Wert. Der optimale Wert bzgl. Antwortzeit und Durchsatz lag zwischen 15 und 20. Dies entspricht einer CNFL-Rate von 16 bis 20 % und einer WKM-Rate von 1.8 bis 2.6.

Bestimmung für den langen Transaktionstyp

Die Abb. 6.8 zeigen die Ergebnisse für den langen Transaktionstyp. Der Fall mit einem Schreibanteil von 10 % ist mit dem 15 %-Fall des kurzen Transaktionstyps vergleichbar. Eine Restriktion des MPL-Wertes unter den Wert von 40 führt zu suboptimalen Ergebnissen. Ab einem MPL-Wert von 25 findet keine wesentliche Veränderung mehr statt.

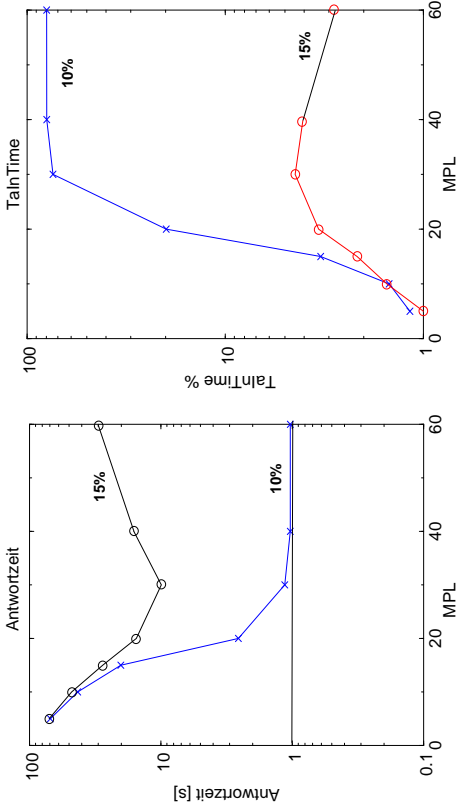


Abb. 6.8: Bestimmung des optimalen SMPL Typ2 - Sperrengpaß 1 Typ

Der optimale statische MPL-Wert liegt für den 15 %-Fall bei 30. Die Antwortzeit wird auf 10 s reduziert. Die TainTime-Anteile erreichen immerhin einen Wert von 5 %. Nach dem optimalen Wert von 30 steigt die Antwortzeit-Kurve wieder steil an.

In Abb. 6.9 wird die Antwortzeit-Zusammensetzung für die beiden Läufe dargestellt. Es ergibt sich ein Bild, daß dem Simulationslauf mit dem kurzen Transaktionstyp ähnelt. Der Sperranteil steigt nur bis zum MPL-Wert von 25 an. Danach bleibt er auf einem konstanten absoluten Niveau stehen. Für den MPL-Wert von 30 beträgt der Sperrbearbeitungsanteil 20 % der gesamten Antwortzeit. Durch einen höheren MPL-Wert wird nur der verbleibende BOT-WS-Anteil weiter vermindert. Für den maximal untersuchten MPL-Wert von 60 steigt dieser Anteil wieder leicht an. Aus den Grafiken läßt sich aufgrund des Maßstabes nicht ersehen, daß die Antwortzeit wieder marginal ansteigt. Der optimale Wert für den MPL liegt somit im Bereich von 40 bis 50.

Die Ergebnisse des langen Transaktionstyps mit 15 % Änderungsanteil entsprechen denen des kurzen Typen mit 25 % Änderungsanteil. Die Sperrwartezeit ist eine streng montone Funktion über dem statischen MPL-Wert. Es existiert ein Minimum bei 30. Im Minimum dauert die Sperrbearbeitung ca. 700 ms bzw. 6 % der gesamten Antwortzeit.

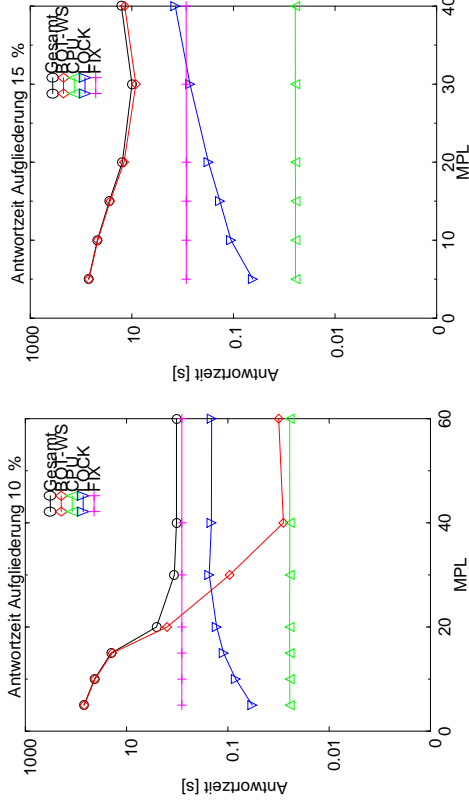


Abb. 6.9: Antwortzeit -Zusammensetzung SMPL, Typ 2, Sperrengpaß 1 Typ

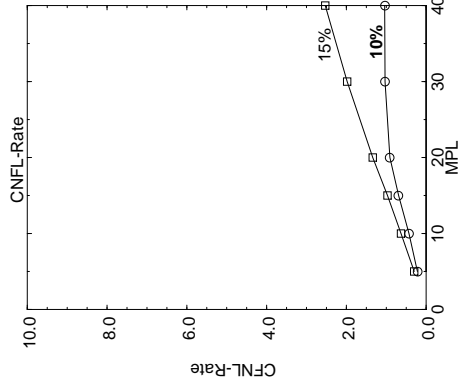


Abb. 6.10: CNFL- und WKM - Rate, SMPL, Typ 2

Auch bei der CNFL- und WKM-Rate ergeben sich Parallelen zum kurzen Referenztyp. Während sich jedoch die WKM-Raten in Bezug auf die optimale statische MPL-Einstellung in gleichen Größenordnungen bewegen, liegen diese Werte für die CNFL-Rate weit auseinander. Für den optimalen MPL-Wert von 30 für den 15 % Fall ist die CNFL-Rate bei 2 % und die WKM-Rate bei 2. Diese Werte lagen beim kurzen Typ bei 18 % und ca. 2.0.

6.1.3 Sensitivitätsanalyse

Im vorigen Abschnitt wurden die optimalen statischen MPL-Werte für zwei verschiedene Typen und zwei verschiedenen Lastsituation bestimmt. Es deutete sich schon bei diesen Simulationen an, daß unter anderem die optimalen Parameter der CNFL- und WKM-Rate für verschiedene Lasten bei optimaler Wahl des statischen MPL-Wertes sich stark unterscheiden können. In diesem Abschnitt werden die drei Parameter Antwortzeitbestandteil, CNFL- und WKM-Rate systematisch variiert. Für die zwei Typen werden mit je zwei verschiedenen Änderungsraten die jeweils besten Parameter für die vorgestellten Regelungsvarianten ermittelt. Bevor mit verschiedenen Transaktionstypen im System operiert wird, sollen diese Experimente Aufschluß darüber geben, ob :

- (1) die Parameter für einen Typ invariant bzgl. der Ankunftsrate sind,
- (2) die Parameter invariant sind bzgl. des Lasttypen,
- (3) es bei mgl. Unterschieden Möglichkeiten gibt, aus den Einzelergebnissen generell gültige Parameter abzuleiten.

Analyse des kurzen Transaktionstyps

Die Abb. 6.11 bis 6.13 zeigen die Ergebnisse für den kurzen Transaktionstyp. Der Sperrbeitragsanteilsprozentsatz wird zwischen den Werten 10 % und 40 % variiert. Die Minima für beide Simulationsläufe liegen an anderen Stellen. Die Einstellung von 15 % ist als guter Kompromißwert anzusehen.

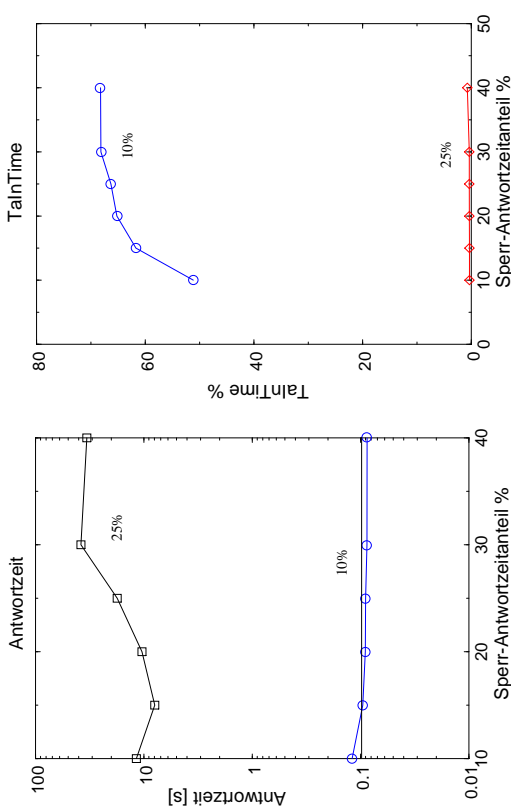


Abb. 6.11: Sperr-Antwortzeitvariation DMPL Regelung Typ I

Die Variation der WKM-Rate wurde für den Wertebereich von 1.1 bis 2.0 vorgenommen. Wie aus Abb. 6.12 zu entnehmen ist, erweist sich für die Steuerung mit der WKM-Rate für den Normal/Hochlastfall von 10 % ein WKM Wert ab 1.2 als guter Wert. Im anderen Szenario ist der WKM-Wert jedoch enger zu fassen. Der optimale WKM Parameter liegt bei 1.5

Der relevante Bereich für die Variation der CNFL-Rate wurde mit dem Intervall von 2 % bis 30 % abgesteckt. Die Konflikttrate muß für den Lauf mit dem geringeren Schreibeanteil mindestens bei 10 % liegen. Für den Lauf mit 25 % Änderungsanteil liegt der optimale Wert bei 20 %.

Der gesamte Verlauf der Antwortzeit findet sich in Abb. 6.13.

Analyse des langen Transaktionstyps

In den Abb. 6.14 bis 6.16 sind die entsprechenden Ergebnisse für den langen Transaktionstyp dokumentiert. Der optimale Antwortzeitanteil liegt bei 20 % für die geringere Änderungsrate und bei 5 % für den Überlastfall. Diese Diskrepanz läßt sich aus den Ergebnissen für die CNFL-Rate ableiten. Die CNFL-Rate liefert für den 15 % Änderungsfall das beste Ergebnis bei einem Wert von 2 %. Die Ergebnisse aus Abb. 6.15 sind interpretationsbedürftig. Die Ergebnisse jenseits einer CNFL-Rate von 15 % sind darauf zurückzuführen, daß der MPL-Wert viel zu hoch ausfällt und die Anzahl der verarbeiteten Transaktionen sehr stark absinkt. Relevant sind folglich nur der Bereich von 2 % bis 10 %. Da die modifizierte Antwortzeit-Erkennung auch auf die CNFL-Rate bei einem MPL-Erhöbungsversuch zurückgreift, ist die eingestellte CNFL-Rate von 10 % kein adäquates Kriterium mehr.

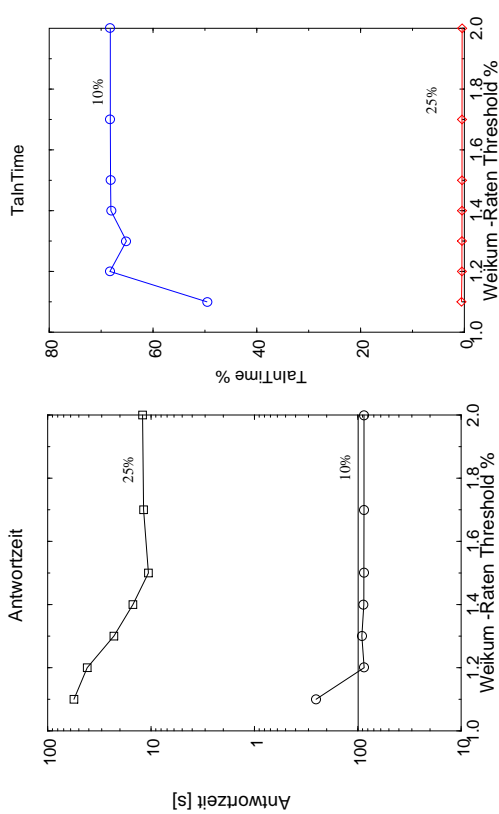


Abb. 6.12: WKM-Raten-Variation, Typ 1

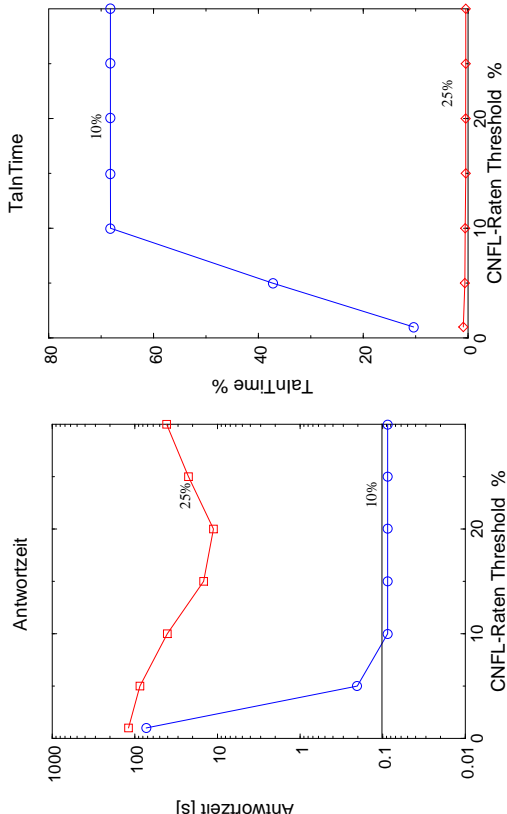


Abb. 6.13: CNFL-Raten-Variation, Typ 1

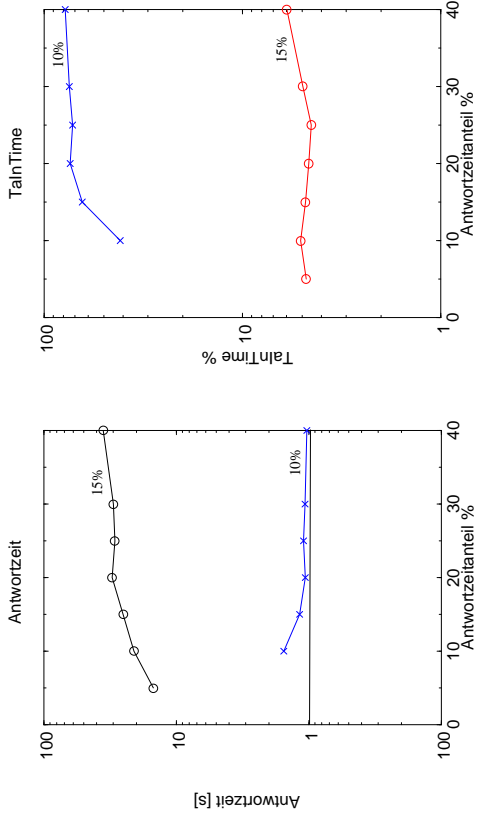


Abb. 6.14: Sperrantwortzeit-Variation, Typ 2

In Abb. 6.16 werden die Ergebnisse für die Steuerung mit der WKM-Rate gezeigt. Gute Einstellungen beginnen hierbei für den Typ 2a ab 1.3 und für den Type 2b liegt er bei 1.8.

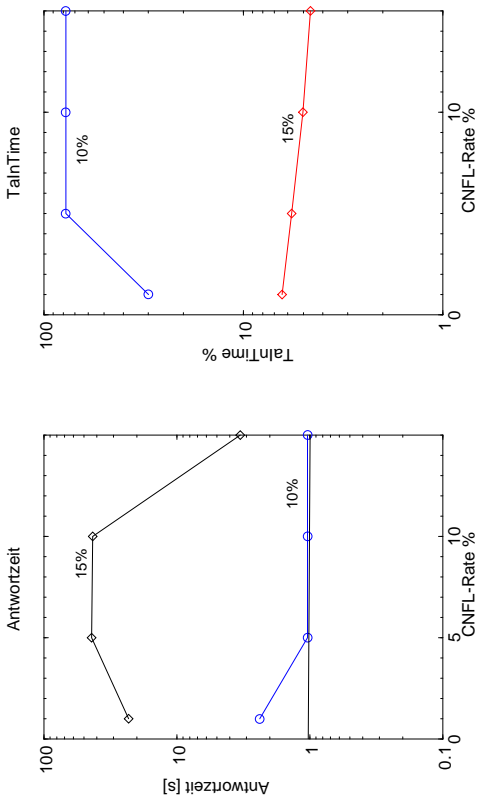


Abb. 6.15: CNFL - Raten-Variation, Typ 2

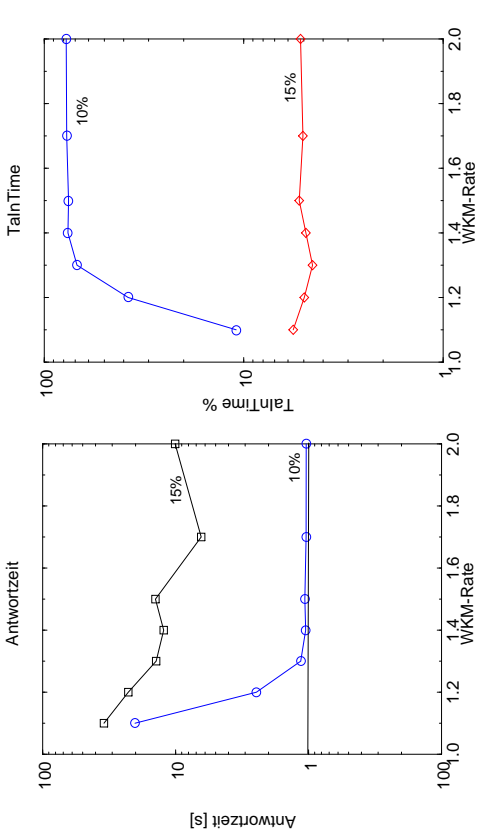


Abb. 6.16: WKM-Raten-Variation, Typ 2

Zusammenfassung der Sensitivitätsanalyse

In der Tabelle 6.3 werden die besten erzielten Antwortzeiten zusammengefaßt. Die Ergebnisse für die Simulationen, die mit einer Einzelsteuerung (individuelle Einstellung des Steuerungsparameters durch Variation) erzielt worden sind, liegen sehr eng zusammen. Es drängt sich aber die Frage auf, ob diese Ergebnisse auch mit einer generellen lastunabhängigen Parametereinstellung erzielt werden kann.

	SMPL [s]	Antwortzeit [s]	CNFL [s]	WKM [s]
5Ref 10 %	0.09	0.09	0.09	0.09
5 Refs 25 %	10	8	10	10
50 Refs 10 %	1	1	1	1
50 Refs 25 %	10	10.5	11	10

Tabelle 6.3: Antwortzeit-Ergebnisse der Sensitivitätsanalyse

In Tabelle 6.4 werden die ermittelten Einstellungen für die Antwortzeitergebnisse aus

Bezeichnung	Referenzen- Änderungsanteil	Antwortzeit	CNFL	WKM
Fall 1a	5 Ref - 10%	>20%	>10%	>1.2
Fall 1b	5 Ref - 25%	15%	20%	1.5
Fall 2a	50 Ref - 10%	>20%	>5%	>1.4
Fall 2b	50 Ref - 15%	5%	2%	1.8

Tabelle 6.4: Parametereinstellungen für optimale Antwortzeit-Ergebnisse

Tabelle 6.3 aufgelistet. Die Fälle mit der niedrigeren Änderungsrate definieren hierbei mit wenigen Ausnahmen die Untergrenze für die Parametereinstellungen. Ausnahmen bilden hier der Fall 1b bzgl. der Antwortzeit und 2b bzgl. der Antwortzeit und CNFL-Rate. Es läßt sich jedoch festhalten, daß die Unterschiede bei den Parametern für **gleiche Lasttypen** mit verschiedenen Ankunftsdaten nicht unwesentlich sind. Die Antwortzeitunterschiede fallen vor allem beim Fall 2b auf. Diese sind jedoch durch die unglückliche Wahl der CNFL-Rate als Begrenzer bei der MPL-Erhöhung zu erklären. Die Überlastfälle sind jedoch nur mit Ein-

schränkungen zu beurteilen. In den Simulationen wird eine Überlast erzeugt, die für die gesamte Simulationszeit andauert¹. Der im Laufe der Simulationsdauer immer größer werdende Antwortzeitanteil für das Warten in der BOT-WS, wirkt sich vor allem auf die Vereinbarkeit der Einstellungsparameter für den Hoch- und Überlastfall aus. Da kein einheitlicher Wert für die Erkennung eines Sperrengpasses gegeben werden kann, ist es um so wichtiger einen richtigen Wert für den Test vor einer MPL-Erhöhung zu finden.

Es ist aber offensichtlich, daß die Konfliktrate typspezifisch vorgegeben werden muß. Da zum augenblicklichen Zeitpunkt dieser Möglichkeit nicht gegeben ist, ist die verwendete Variante schon an dieser Stelle sehr kritisch zu beurteilen.

Recht nahe bei einander liegen die WKM-Werte. Dies ist vor allem auch deshalb hervorzuheben, da selbst der Fall 1b mit einem WKM-Wert von 1.8 nur unwesentlich schlechter liegt als mit dem optimalen Wert (vgl. Abb. 6.12).

Die durchgeführten Simulationen lieferten bisher kein einheitliches Ergebnis, die die Vorgabe oder Kombination von verschiedenen Parametern für die weiteren Untersuchungen reifen würde. Die einzelnen Parameter liegen zum Teil in unmittelbarer Nachbarschaft; ein allgemeiner, bzgl. Lasttypen und Ankunftsrate invarianten Parameter, kann jedoch nicht abgeleitet werden.

1. Die Länge der BOT-WS nimmt direkt proportional zur vergangenen Simulationszeit zu.

6.2 Simulationen mit zwei Transaktionstypen

In diesem Abschnitt sollen die bisher ermittelten Ergebnisse anhand zweier synthetischer Lasten überprüft werden. Das Vorgehen verläuft analog zu den letzten Experimenten. Die Experimente betrachten dabei sowohl den Fall mit konstanten Ankunftsverhalten (in Abschnitt 6.2.1) als auch den Fall mit Lastspitzen (in Abschnitt 6.2.2). Im folgenden Abschnitt 6.4 werden die so gewonnenen Erkenntnisse mit denen aus der Beobachtung des Regelungsverfahrens bei einer realen Last verglichen.

6.2.1 Simulationen mit konstantem Ankunftsverhalten

Die aus den letzten Abschnitten bekannten Transaktionstypen werden in diesen Simulationen leicht variiert und zu einer Last zusammengefaßt. Der Transaktionstyp 1 wird als kurze Änderungsaktion aufgefaßt. Die Änderungsrate wurde auf 9 % gesetzt. Die mit 50 Referenzen im Vergleich lange Transaktion ist eine reine Lesetransaktion. Die Ankunftsrate der Transaktionen wurde auf 175 tps für Typ 1 und 20 tps für Typ 2 eingestellt. Das Verhältnis der Ankunftsrate beträgt somit ungefähr 9:1. Bezieht man die verschiedene Referenzanzahl ein, so erhält man ein Verhältnis von ca. 1.1 : 1. Die Transaktionen greifen auf beide Partitionen zu. Da die Partitionsgröße der ersten Partition mit 100 Seiten relativ klein ist, wurde die Referenzwahrscheinlichkeit für Transaktionsyp 1 für diese Partition auf 10 % gesetzt. Die Zielantwortzeiten wurden auf 150 ms für den Typ 1 und 1000 ms für den Typ 2 gesetzt. Die für die MPL-Erhöhung beim antwortzeitbezogenen Verfahren relevante CNFL-Rate wurde auf 5 % gesetzt. Der Korridor wurde auf 1 % eingestellt. Diese Werte gelten sowohl für die GMPL- als auch für die LMPL-Regelung. Die anderen relevanten Parameter finden sich in Tabelle 6.11.

Die Wahl der oben beschriebenen Parameter für die Lastgenerierung und die Systemumgebung führen zu einer Hochlast. Lediglich im Fall einer sehr hohen MPL-Einstellung wandelt sich diese zur Überlast. Die *Überforderung* des Systems liegt aber in einer anderen Größenordnung als dies bei den Überlastfällen der Simulation mit nur einem Transaktionstyp der Fall war.

In keinem der analysierten Fälle spielte die Rücksetzquote aufgrund von Deadlocks eine wesentliche Rolle. Deadlocks und die damit verbundenen Rücksetzungen traten nur sporadisch auf.

Parameter	Partition 1 / Transaktionstyp 1	Partition 2/ Transaktionstyp 2
Relation		
Anzahl Partitionen	2	
Anzahl Objekte	1.000	10.0000
Blockungsfaktor	10	10
Transaktionen		
Anzahl	2	
Referenzen	5	50
Zielantwortzeiten [s]	0.15	1.0
Schreibanteil [%]	9	0
Zugriffsverteilung (P1,P2) [%]	(0.5,0.5)	(0.1,0.9)
Sperrgranulat	Objekte	
Sperrmethode	2-phasiges Sperren	
Puffergröße	2.000	
Logging	JA	
CPU		
Anzahl	2	
Leistung (je Prozessor)	15 MIPS	

Tabelle 6.5: Grundparameter der Sperr-Engpaß-Simulationen mit 2 Typen

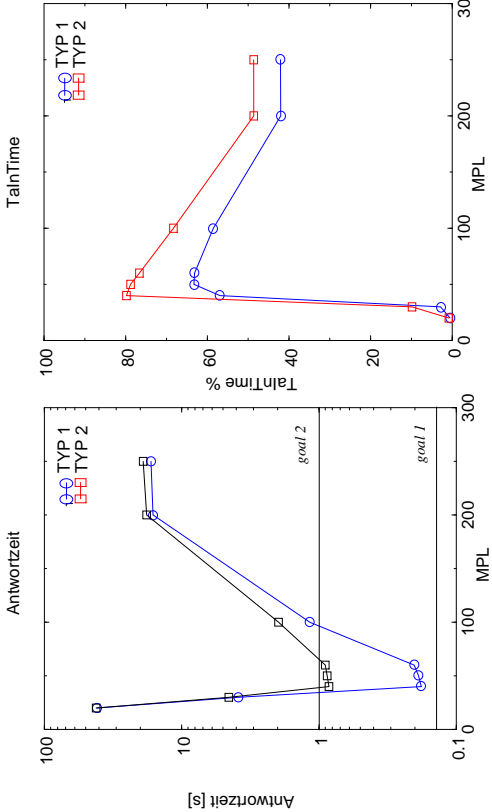


Abb. 6.17: Bestimmung des optimalen statischen MPL-Wertes, 2 Typen-Sperrengpaß

In Abb. 6.17 sind die Antwortzeit-Ergebnisse für statische MPL-Einstellungen dargestellt. Es zeigt sich, daß der MPL-Wert sowohl zu niedrig als auch zu hoch eingestellt werden kann. Der Transaktionstyp 2 erreicht für gute MPL-Einstellungen im Durchschnitt sein Ziel. Die TaInTime-Anteile liegen dennoch nur bei max. 80 %. Obwohl der Transaktionstyp 1 die vorgegebene Zielantwortzeit in keinem Fall erreicht, liegen die TaInTime-Anteile mit über 60 % für gute Einstellungen sehr hoch. Eine *guter* MPL-Wert liegt für die gemischte Last im Intervall von 40 bis 55.

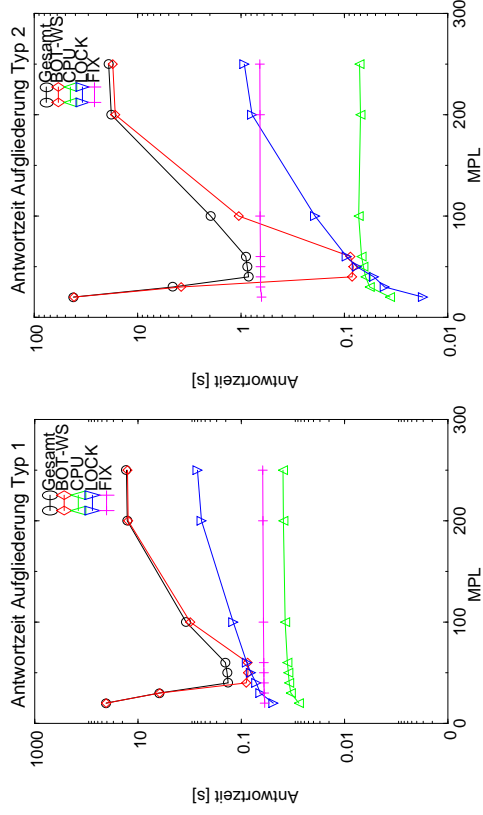


Abb. 6.18: Antwortzeit-Aufgliederung, SMPL-Einstellung

In Abb. 6.18 werden die Antwortzeitbestandteile der beiden TA-Typen für die statische MPL-Einstellung gezeigt. Beim Transaktionstyp 1 sind die Antwortzeitbestandteile für die E/A- und die CPU-Bearbeitung fast konstant. Wesentliche Änderung sind nur beim Sperrbearbeitungs- und BOT-WS-Anteil zu erkennen. Im für die Antwortzeit optimalen Bereich liegt der absolute Antwortzeitanteil für die Sperrbearbeitung bei ca. 65 ms oder etwa 34 % der gesamten Antwortzeit von 190 ms. Dies entspricht 45 % der gesamten Antwortzeit. Beim langen Transaktionstyp ist die Situation ähnlich. Der Anteil für die E/A- und CPU-Bearbeitung ist konstant. Der Sperrbearbeitungsanteil liegt bei ca. 80 ms. Dies entspricht 10 % der gesamten Antwortzeit für den MPL-Wert von 40. Obwohl die Sperrbearbeitungsanteilskurve stark ansteigt, vermindert sich dieses relative Verhältnis leicht. Für den schlechtesten Fall mit einem statischen MPL-Wert von 250 liegt die Antwortzeit bei 11 s und die Sperrbearbeitung bei 0.9 s. Dies entspricht lediglich einer Quote von 8 %.

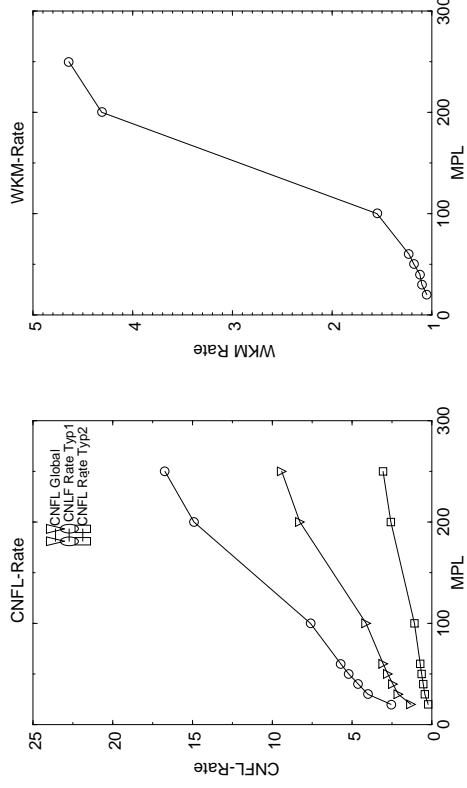


Abb. 6.19: CNFL- und WKM-Rate SMPL-Einstellung - Sperrengpaß 2 Typen

In Abb. 6.19 werden die über der Simulationszeitraum gemittelten Werte für die CNFL-Raten und die WKM-Rate dargestellt. Die CNFL-Raten werden sowohl typespezifisch als auch global ermittelt. Die CNFL-Rate für den kurzen Typ liegen zwischen den Werten 2.5 % und 17 %. Für den optimalen Bereich der MPL-Einstellung von 40 bis 50 liegt die Quote zwischen 3 % und 4 %. Die CNFL-Rate für den zweiten Typ liegt bei optimaler Einstellung des MPL-Wertes bei nur 1 %. Der über alle Transaktionen ermittelte globale Wert liegt bei 2.5 %.

Die WKM-Rate liegt für den relevanten Bereich zwischen 1.1 und 1.2. Der steile Anstieg der Werte ab einem MPL-Wert von 100 erklärt die Antwortzeitentwicklungen in Abb. 6.17 und 6.18.

In den Abb. 6.20 bis 6.22 sind die Ergebnisse für die Variation der drei maßgeblichen Einflußparameter abgebildet. Für die reine Steuerung mit CNFL-Rate und WKM-Rate lassen sich optimale Intervalle für die Steuerung angeben. Die antwortzeitbezogene Erkennung liefert um so bessere Ergebnisse je höher der erlaubte Antwortzeitanteil eingestellt ist. Diese Beobachtung gilt bis zum Wert von 60 %. Ein Sperrengpaß wird für diesen Wert nur noch sehr selten erkannt. Der MPL-Wert kann trotz der Einschränkung durch die Konfliktquote einen zu großen Wert annehmen.

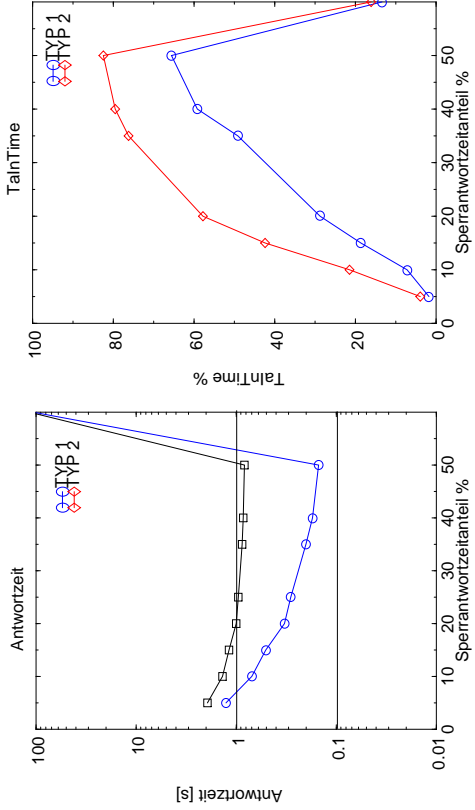


Abb. 6.20: Antwortzeitanteil-Variation, Sperrkonflikt, 2 Typen

Die Entwicklung der Antwortzeit bis zum Anteilswert von 50 % war in dieser Form nicht vorhersehbar. Für niedrigere Werte werden zu häufig Sperrengpässe erkannt und dementsprechend auch MPL-Reduktionen vorgenommen. Je größer der Anteilswert ist, desto geringer ist auch die Häufigkeit der MPL-Reduktion. Ein Anteilswert von 50 % ist jedoch nicht als wünschenswert anzusehen. Einen Engpaß erst ab einem 50 % Anteil an der gesamten Antwortzeit zu erkennen, widerstrebt den empirischen Erfahrungen und der Anschauung. Das Gegenstück zur MPL-Reduktion durch die Engpaßerkennung ist die BOT-WS-Erkennung. Es ist also an dieser Stelle zu hinterfragen, ob nicht eine zu geringe Häufigkeit der MPL-Erhöhung insgesamt zu einem zu niedrigen durchschnittlichen MPL-Wert führt. Da die BOT-WS Wartezeit auf 10 % der gesamten Antwortzeit gesetzt wurde, kann man davon ausgehen, daß die zusätzliche Bedingung der CNFL-Rate Einhaltung der maßgebliche Faktor für eine mögliche Anpassung ist. Die CNFL-Rate wurde in diesem Fall auf 5 % gesetzt und der Korridor lag bei 1 %. Dies bedeutet, daß eine MPL-Erhöhung nur dann gestattet ist wenn die CNFL-Rate unter 4,95 % liegt. Im Vergleich zu den durchschnittlichen CNFL-Raten bei der statischen MPL-Einstellung scheint dieser Wert zu hoch gewählt. Es ist aber zu bedenken, daß die Werte aus

Abb. 6.19 Mittelwert darstellen.

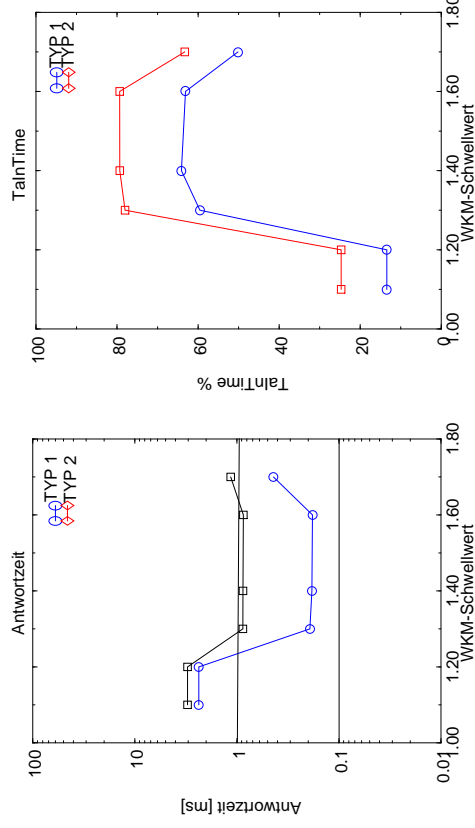


Abb. 6.21: WKM - Raten-Variation, Sperrkonflikt, 2 Typen

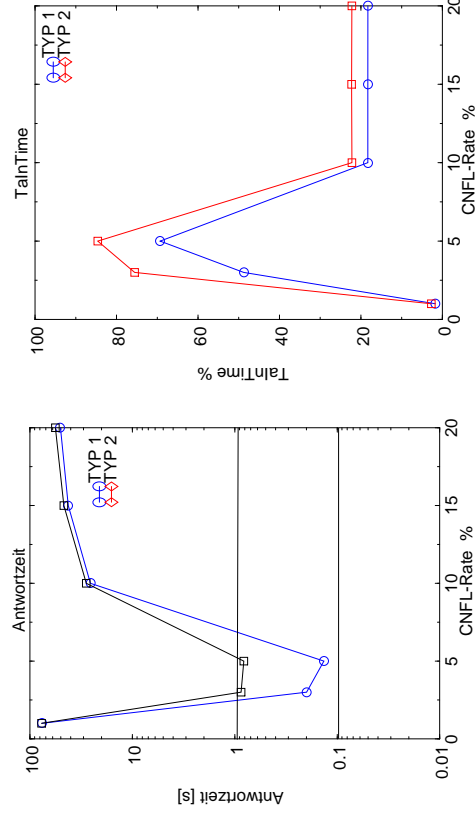


Abb. 6.22: CNFL-Rate-Variation, Sperrkonflikt, 2 Typen

Die Ergebnisse der Simulationen mit reiner Steuerung durch die WKM-Rate werden in Abb. 6.21 gezeigt. Das Intervall der optimalen Ergebnisse liegt für diese Größe zwischen 1.25 und 1.6. Die Ergebnisse sind für diesen Bereich als gleichwertig anzusehen.

Die Ergebnisse für die CNFL-Raten-Variation sind in Abb. 6.22 dokumentiert. Die besten Ergebnisse liefert hier der Wert von 5 %. Der Bereich für gute Ergebnisse liegt hierbei zwischen 2.5 % und ca. 7 %.

Verfahren	TYP 1 Antwortzeit [ms]	TaIn Time [%]	TYP2 Antwortzeit [ms]	TaIn Time [%]
<i>SMPL</i>	188	63	871	78
<i>DMPL</i>	150	66	828	82
<i>WKM</i>	148	67	894	83
<i>CNFL</i>	136	69	817	84

Tabelle 6.6: Optimale Antwortzeitergebnisse, globale MPL-Regelung

In der Tabelle 6.6 werden die besten Antwortzeitergebnisse für die globale MPL-Regelung zusammengefaßt. Die besten Ergebnisse liefert die Regelung mit der CNFL-Rate. Die beiden anderen Verfahren sind dicht bei einander in Bezug auf den ersten Typ. Bei der WKM-Rate ist die Antwortzeit des zweiten Typs um ca. 60 ms schlechter. Die Einschränkungen der Ergebnisse des antwortzeitbezogenen Verfahren wurden bereits in der Diskussion der Einzelergebnisse erläutert. Es läßt sich festhalten, daß im optimalen Fall die dynamische Regelung des globalen MPL-Wertes bessere Ergebnisse als die statische MPL-Einstellung liefert. Obwohl man dieser Feststellung entgegenhalten kann, daß die statische MPL-Schrittweite von 10 bei der Bestimmung des optimalen MPL-Wertes zu grob ist, ist dies im Vergleich zu den Ergebnissen bei der CPU-Engpaßbekenennung ein bemerkenswertes Resultat.

In der Tabelle 6.7 sind die durchschnittlichen MPL-Werte und die jeweils eigenstellten Parameter für die Ergebnisse aus Tabelle 6.6 aufgelistet. Die durchschnittlichen MPL-Werte liegen alle über dem besten statischen Wert.

Verfahren	Durchschnittlicher MPL-Wert	Eingestellter Parameter
<i>SMPL</i>	40	
<i>DMPL</i>	51	50%
<i>WKM</i>	50	1.5
<i>CNFL</i>	56	5%

Tabelle 6.7: Durchschnittliche MPL-Werte und Parametereinstellungen

In den Abb. 6.23 bis 6.25 sind die Ergebnisse für die LMPL-Regelung dokumentiert. Beim lokalen Verfahren kann die Erkennungs- und Regelungsstrategie zwei verschiedene Richtungen einschlagen. Wenn in der vorliegenden Konstellation beide Transaktionstypen ihre Antwortzeit verfehlen, so wird ein globaler Engpaß erkannt. Die Regelung verläuft dann analog zum globalen MPL-Regelungsverfahren.

Verfehlt im 2-Typen-Fall nur eine Klasse ihre Antwortzeit, so wird ein lokaler Engpaß erkannt. Die Regelung des MPL-Wertes verläuft mit einem Unterschied analog zur Regelung des globalen Wertes. Die Regelung betrifft nur den lokalen MPL, der für die kritische Transaktionsklasse verwendet wird.

Bei einem Sperrenpaß wird eine Ergänzung des lokalen MPL-Konzeptes vorgenommen. Tritt ein Sperrenpaß auf, so wird der lokale MPL-Wert der kritischen Transaktionsklasse nur dann verringert, wenn die Transaktionen der eigenen Klasse auch der Hauptursacher des Engpasses sind. Ist dies nicht der Fall, so wird der **gemischte MPL-Wert** verringert. Dieser Wert legt fest, wie viele Transaktionen zweier verschiedener Klassen gleichzeitig verarbeitet werden dürfen. Dieses Konzept kommt in den Simulationen nur sehr eingeschränkt zum Tragen. Die verwendete Implementation initialisiert die gemischten MPL-Werte jeweils mit der Summe der lokalen MPL-Wert. Bei **jeder** Veränderung eines lokalen MPL-Wertes wird der gemischte MPL-Wert wieder initialisiert. Die zeitliche Bedeutung eines gemischten MPL-Wertes ist folglich sehr eingeschränkt.

In den Simulationsläufen wurden mit Regelungen, die auf Antwortzeitanteilen, CNFL-Raten und WKM-Raten basieren, experimentiert. Bei Erkennung einen lokalen Problems werden die Antwortzeitbestandteile und die CNFL-Rate mit den Werten der jeweils betrachteten Transaktionsklasse verglichen. Die global vorgegebene CNFL-Rate wird mit den für jeden Transaktionstyp (i) berechneten CNFL-Rate (i) verglichen.

Die Verwendung der WKM-Rate als Steuerungsparameter für die lokale MPL-Regelung stellt einen konzeptionellen Bruch dar. Die WKM-Rate ist ein **globaler Wert**. Ihre Verwendung auf lokaler Ebene ist deshalb nicht offensichtlich. Die Simulationen werden der Vollständigkeit halber dennoch durchgeführt.

Die Bestimmung einer optimalen statischen Einstellung mit lokalen MPL-Werten wurde nicht vorgenommen, da bereits bei zwei zu variierenden Parametern, die Anzahl der durchzuführenden Simulation sehr groß ist.

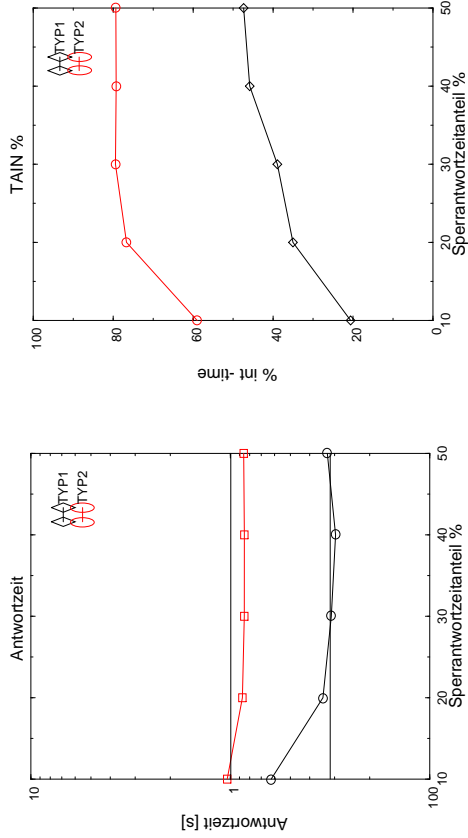


Abb. 6.23: LMPL-Regelung, Antwortzeit-Variation, Sperrkonflikte 2 Typen

Die antwortzeitorientierte Regelung liefert für die Einstellung des Sperrbearbeitungsanteils von ca. 25 % bis 40 % gute Ergebnisse. Das beste Ergebnis wird für die Einstellung von 40 % erreicht. Die Antwortzeit beträgt bei dieser Einstellung 296 ms für Typ 1 und 850 ms für Typ 2. Mit diesen durchschnittlichen Antwortzeiten werden TainTime-Anteile von 45 % bzw. 79 % erreicht. Die durchschnittlichen MPL-Werte variieren von ca. 39 für die 10 % Einstellung bis zu einem Wert von 46 für die Einstellung von 50 %. Die antwortzeitbezogene Erkennung liefert für einen relativ breiten Intervall des Sperrbearbeitungsanteils ein gutes Ergebnis. Für die betrachteten Werte trat kein Durchsatzproblem auf.

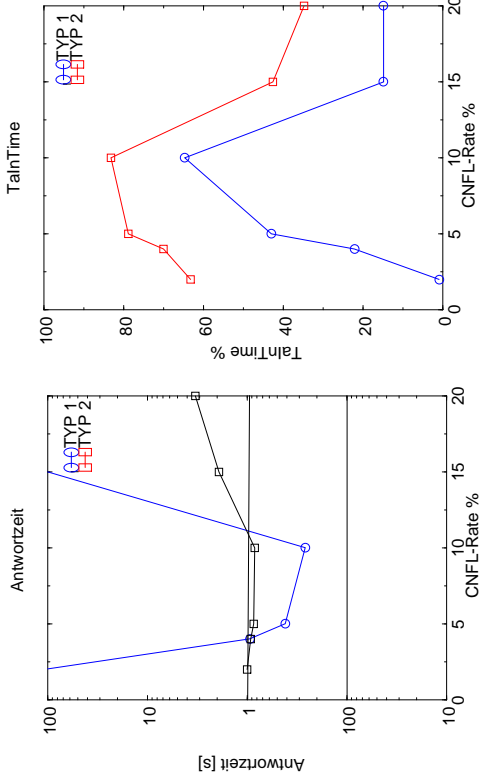


Abb. 6.24: LMPL-Regelung, CNFL-Raten-Variation, Sperrkonflikte 2 Typen

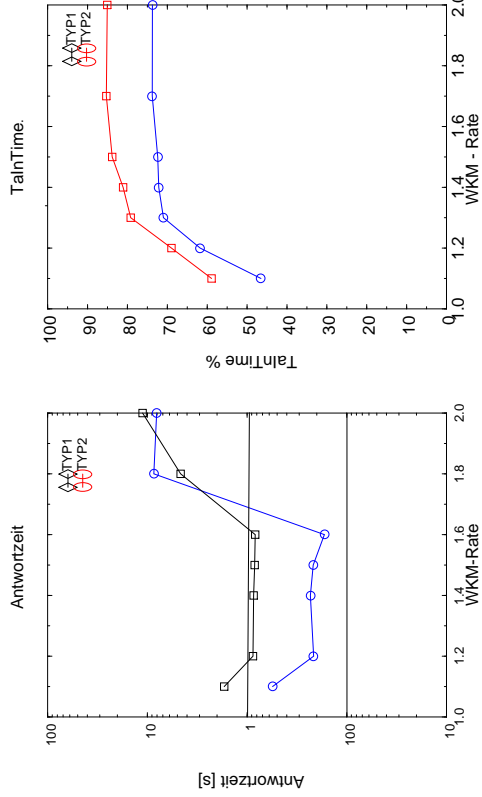


Abb. 6.25: LMPL-Regelung, WKM-Raten-Variation, Sperrkonflikte 2 Typen

Die Ergebnisse für die Steuerung mit einer einheitlichen CNFL-Rate werden in Abb. 6.24 gezeigt. Die Wahl der CNFL-Raten beschränkt sich auf ein kleines Intervall von 5 % bis max. 10 %. Bereits ab einem Wert von 15 % treten neben der Antwortzeitverschlechterung auch erhebliche Durchsatzprobleme auf.

Abb. 6.25 stellt die Ergebnisse des Simulationslaufes mit der WKM-Rate dar. Die Ergebnisse sind bei Einstellungen zwischen 1.2 und 1.6 stabil. Ab einer WKM-Rate von 1.8 verschlechtern sich die Antwortzeitergebnisse erheblich. Die Antwortzeiten erreichen Werte von 8600 ms für Typ 1 und 4640 ms für Typ 2. Ab diesem Schwellwert konnte eine Verminderung des Durchsatzes festgestellt werde. Für den relevanten Bereich von 1.2 bis 1.6 liegen die über dem Simulationszeitraum gemittelten MPL- Werte zwischen ca. 30 und 68.

In der Tabelle 6.6 werden die besten Ergebnisse für die lokale MPL-Regelung zusammengefaßt.

Verfahren	TYP 1 Antwortzeit [ms]	TaInTime [%]	TYP2 Antwortzeit [ms]	TaInTime [%]
<i>SMPL (global)</i>	188	63	871	78
<i>Antwortzeit</i>	296	45	850	79
<i>CNFL</i>	261	64	840	83
<i>WKM</i>	167	68	829	84

Tabelle 6.8: Optimale Antwortzeitergebnisse, LMPL-Regelung

Die Ergebnisse zeigen, daß die lokale MPL-Regelung den **Transaktionstyp 1 deutlich benachteiligt**. Für den Typ 1 werden, mit Ausnahme der Steuerung mit globaler WKM-Rate, in keinem Fall die Antwortzeiten der globalen statischen MPL-Einstellung übertroffen. Die Ergebnisse für den zweiten Typ sind besser als beim statischen Verfahren. Dies läßt sich auch an Hand der Tabelle 6.7 nachvollziehen. Die durchschnittlichen globalen MPL- Werte sind alle höher als beim besten statischen Simulationslauf. Die lokalen MPL-Werte zeigen jedoch, daß zwischen der Ankunftsrate und dem lokalen MPL-Wert für den Transaktionstyp 1 eine relativ große Differenz besteht.

Verfahren	Durchschnittlicher globaler MPL-Wert	lokaler MPL Typ1	lokaler MPL Typ2	Eingestellter Parameter
<i>SMPL</i>	40			
<i>Antwortzeit</i>	44	23	20	40%
<i>CNFL</i>	68	43	25	10%
<i>WKM</i>	63	41	22	1.6

Tabelle 6.9: Durchschnittliche MPL-Werte und Parametereinstellungen

Die Ergebnisse der lokalen MPL Regelung sind im Vergleich zur Regelung mit dem globalen MPL somit als schlechter zu bewerten. In Bezug auf die TaInTime-Anteile sind die Ergebnisse jedoch sehr dicht bei einander. Nur die lokale MPL-Regelung mit antwortzeitorientierter Erkennung weicht negativ von dieser Beobachtung ab. Die TaInTime-Anteile sind für die übrigen Fälle besser als die besten statischen Ergebnisse.

6.2.2 Simulationen mit stark variierender Last

Aufbauend auf den Simulationen aus dem vorigen Abschnitt, wurde ein Simulationsszenario mit Lastspitzen entworfen. Die Transaktionstypen und ihr Referenzverhalten wurden beibehalten. Der Schreibanteil des Transaktionstyps 1 wurde auf 7 % gesenkt. Die Ankunftsraten betragen 175/20 tps (Typ 1/Typ 2) im Normallastfall und 200/20 tps im Hochlastfall. Die Zielantwortzeiten wurden den neuen Verhältnissen angepaßt und auf Werte von 80 ms bzw. 900 ms gesetzt. Über Beginn und Dauer der Lastspitzen gibt die Tabelle 6.10 Aufschluß. Die Simulationsdauer beträgt 500 s.

Nummer	1	2	3	4	5
Beginn	50	150	265	340	375
Dauer	20	15	20	15	20

Tabelle 6.10: Die Lastspitzenverteilung

Die meisten Simulationsparameter wurden aus den Simulationen mit stabiler Last übernommen. In Tabelle 6.11 werden die veränderten Einstellungen zusammengefaßt.

Parameter	Transaktionstyp 1	Transaktionstyp 2
Transaktionen		
Anzahl	2	
Referenzen	5	50
Ankunftsrate (Normal/Peak)	175/200	20/20
Zielantwortzeiten [s]	0.08	0.9
Schreibanteil [%]	7	0
Zugriffsverteilung (P1,P2) [%]	(0.5,0.5)	(0.1,0.9)

Tabelle 6.11: Grundparameter der Sperr-Engpaß-Simulationen mit 2 Typen, Variierende Last

Die Simulationsabfolge entspricht der aus den letzten Abschnitten bekannten Vorgehensweise. Die Simulationen aus dem letztem Abschnitt haben jedoch gezeigt, daß beim antwortzeitbezogenen Verfahren mit mehr als einem Transaktionstyp die CNFL-Rate als globale Größe zur Steuerung nur sehr eingeschränkt verwendbar ist. Aus diesem Grunde wurde eine

zusätzliche Simulationsreihe angelegt. An die Stelle der CNFL-Rate wird beim antwortzeitbezogenen Verfahren die WKM-Rate vor der MPL-Erhöhung abgeprüft. Die Simulationen aus dem letzten Abschnitt haben gezeigt, daß die WKM-Rate weitaus unabhängiger vom Transaktionstyp und der Lastsituation gewählt werden kann. Der Schwellwert für die WKM-Rate wird auf 1.5 und der Korridor auf 10 % gesetzt. Die Simulationsreihen bei der antwortzeitbezogenen Erkennung werden im folgenden in Kurzform als Antwortzeit I (Verwendung der CNFL-Rate) und Antwortzeit II (Verwendung der WKM-Rate) verwendet. Von der Verwendung der WKM-Rate bei lokaler MPL-Regelung wird Abstand genommen, da die WKM-Rate nur Informationen über den globalen Systemzustand wiedergibt.

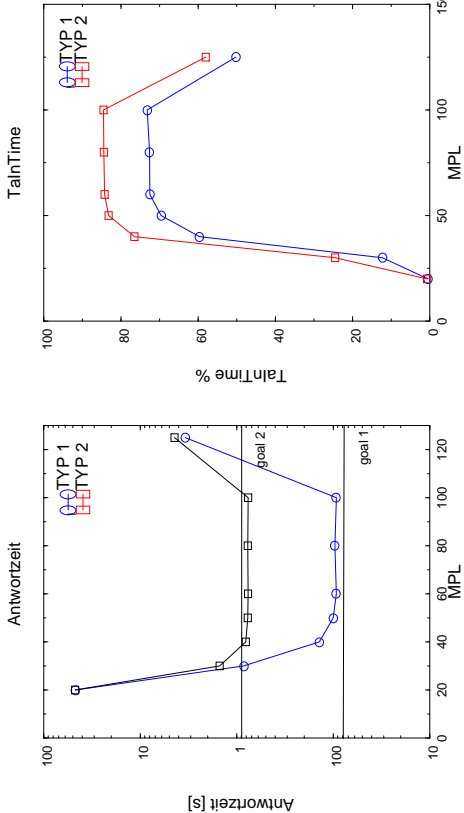


Abb. 6.26: SMPL-Einstellung, Sperrkonflikte 2 Typen, Variierende Last

Die Abb. 6.26 zeigt den Antwortzeitverlauf bei statischer MPL-Einstellung. Die Wahl des MPL-Wertes ist relativ unkritisch zu nennen, da die Antwortzeit-Werte im Intervall von 50 bis 100 als konstant anzusehen sind. Dies gilt auch für die TaktTime-Anteile. Der MPL-Wert kann auch zu hoch gewählt werden, da für den MPL-Wert von 120 eine wesentliche Verschlechterung der Ergebnisse eintritt.

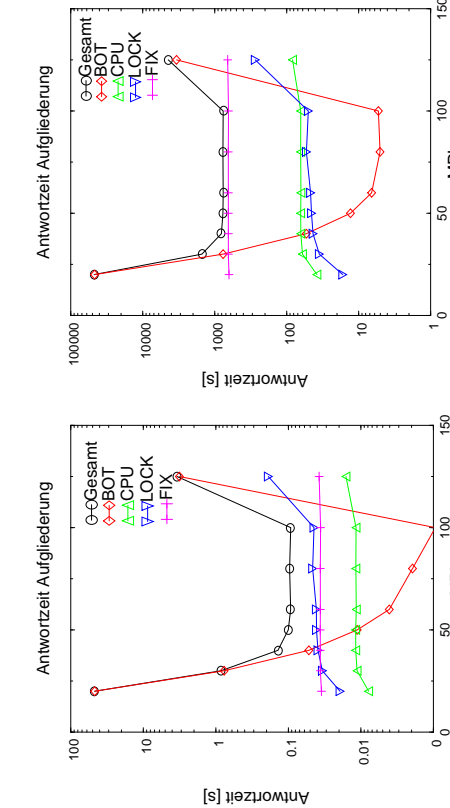


Abb. 6.27: SMPL-Einstellung, Sperrkonflikte 2 Typen, Variierende Last

Die Aufgliederung der Antwortzeit für die beiden Transaktionstypen in Abb. 6.27 kann für die Last mit Lastspitzen nur eingeschränkt Informationen über die Simulationsläufe geben. Die Werte stellen Mittelwerte über dem Simulationszeitraum dar. Bei den Simulationen aus dem letzten Abschnitt kann man von einem relativ homogenen Ankunftsverhalten ausgehen. Dies ist bei der Simulationen aus diesem Abschnitt nicht der Fall.

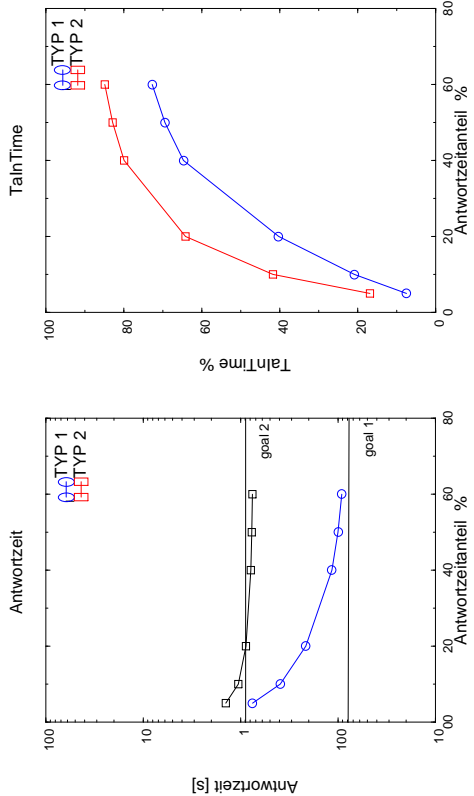


Abb. 6.28: GMPL-Regelung, Antwortzeit-Erkennung I, Sperrkonflikte 2-Typen , Variierende Last

Wie auch im Fall mit stabiler Last kann der Antwortzeitanteil lediglich zu gering gewählt werden. Eine MPL-Erhöhung wird bei den Simulationen nur bei einer CNFL-Rate von unter 5 % erlaubt.

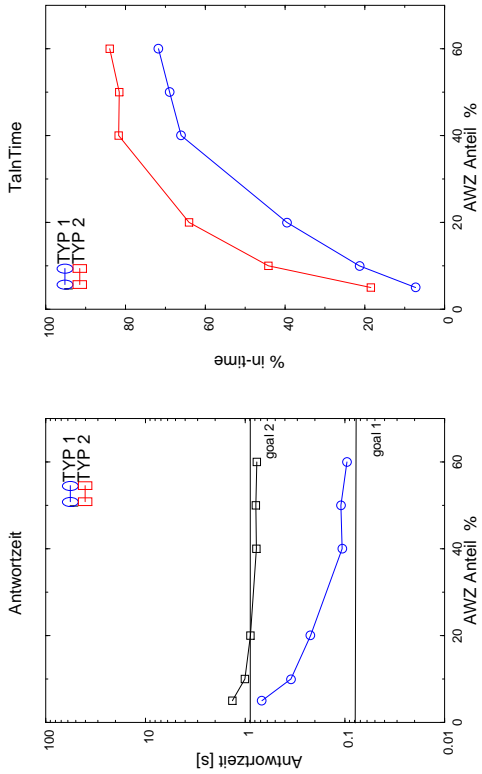


Abb. 6.29: GMPL-Regelung, Antwortzeit-Erkennung II, Sperrkonflikt, 2-Typen , Variierende Last

Abb. 6.29 zeigt die Ergebnisse für das antwortzeitbezogene Verfahren mit Verwendung der WKM-Rate. Die Unterschiede zwischen den beiden antwortzeitbezogenen Verfahren sind bei variabler Last nur marginal.

Abb. 6.30 zeigt die Ergebnisse für die Steuerung mit CNFL-Rate. Die Ergebnisse mit stabiler Last haben eine quasi analogen Verlauf wie die Ergebnisse hier. Das beste Ergebnis liefert eine CNFL-Rate von 5 %.

Abb. 6.31 dokumentiert die Ergebnisse für die Steuerung mit Hilfe der WKM-Rate. Den WKM-Raten-Wert kann für die betrachteten Werte lediglich zu gering gewählt werden. Die weicht von den Beobachtungen bei stabiler Last ab. Dort wurde ab einer WKM-Rate von 1.7 das Ergebnis schlechter. Der maximal betrachtete Wert liegt bei den Simulationen mit stark variierender Last bei 2.0. Es ist möglich, daß bei einer höheren Einstellung des Wertes ein ähnliches Verhalten auftritt. Da jedoch ab einem MPL-Wert von 1.3 weder die TaktTime-Anteile als auch die Antwortzeit sich kaum verändert, kann der WKM-Wert aus dem Intervall von 1.3 bis 2.0 gewählt werden.

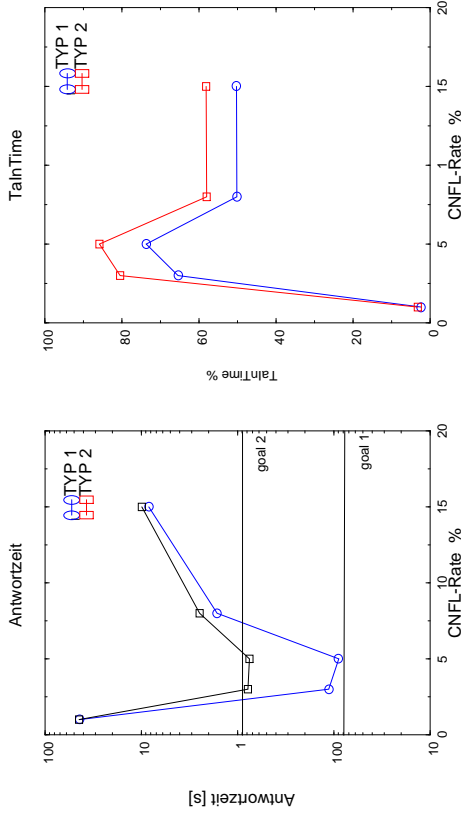


Abb. 6.30: GMPL-Regelung, CNFL-Raten-Variation, Sperrkonflikte 2-Typen, Variierende Last

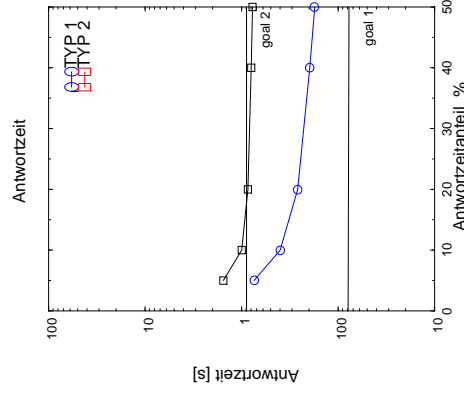


Abb. 6.32: LMPL-Regelung, Antwortezeitanteil-Variation, Sperrkonflikte 2-Typen, Variierende Last

In den Abb. 6.32 bis 6.34 sind die Ergebnisse für die Simulationsläufe mit lokaler MPL-Regelung dargestellt.

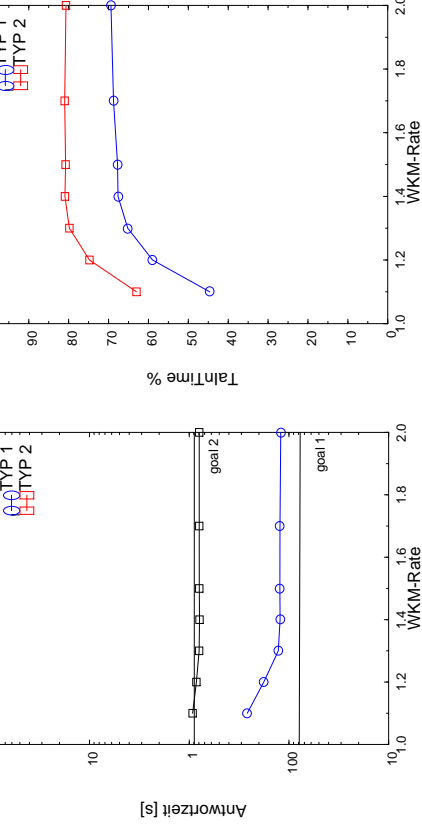
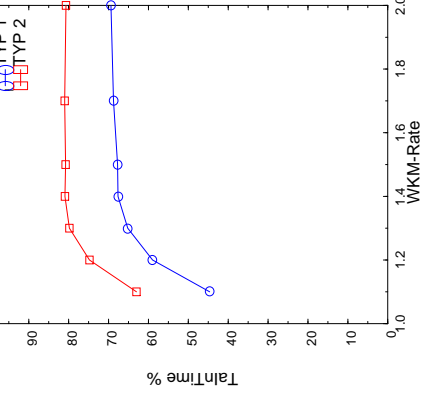


Abb. 6.31: GMPL-Regelung, WKM-Raten-Variation, Sperrkonflikte 2-Typen, Variierende Last



Bei der antwortzeitbezogenen Erkennung ergibt sich ein ähnliches Bild, wie bei der entsprechenden globalen MPL-Regelung. Der MPL-Wert kann nur zu gering gewählt werden. Das beste Antwortzeitergebnis wird für die Wahl von 50 % für den Sperrbearbeitungsprozentsatz erhalten. Die durchschnittliche Antwortzeit für den Transaktionstyp 2 liegt unter den Zielvorgaben. Die durchschnittlichen Antwortzeiten des Transaktionstyp 1 liegen deutlich über den Ergebnissen bei globaler MPL-Regelung.

Die beste Einstellung für die Regelung mittels der CNFL-Rate wird für einen Wert von 8 % erhalten. Aber bereits für den CNFL-Raten-Wert von 19 % verschlechtert sich das Ergebnis drastisch. Dies entspricht u.a. auch den Ergebnissen mit globaler Regelung. Obwohl die Variation der CNFL-Rate auf dem begrenzten Intervall von 1 % bis 10 % vorgenommen wurde, ergibt sich für den Transaktionstyp 1 ein optimales Ergebnis nur für einen Wert. Während der Antwortzeit-Verlauf für den langen Transaktionstyp relativ gleichmäßig über dem gesamten Intervall verläuft, ergeben sich für den kurzen Transaktionstyp sehr starke Veränderungen.

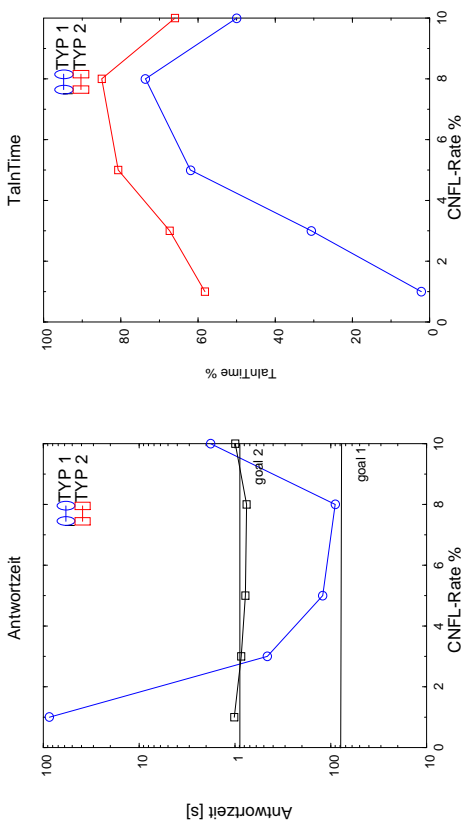


Abb. 6.33: LMPL-Regelung, CNFL-Raten-Variation
Sperrkonflikte 2-Typen, Variierende Last

In Tabelle 6.6 werden die Ergebnisse noch einmal zusammengefaßt. Die Ergebnisse fallen für die beiden betrachteten Verfahren mit globaler und lokaler MPL-Wert-Regelung durchaus unterschiedlich aus.

Bei der globalen MPL-Regelung liegt lediglich das Verfahren mit der WKM-Raten-Regelung über dem statischen MPL-Wert-Ergebnissen. Sowohl die Regelung über die Antwortzeitan-teile als auch die CNFL-Raten orientierte Steuerung liefern Ergebnisse, die einige Millisekun-den unter denen des statischen Verfahrens mit optimaler Einstellung liegt. Die Einstellung der CNFL-Rate mit 5 % für die MPL-Erhöhung war unter den gegebenen Voraussetzungen sehr gut. Dies zeigt sich auch bei der CNFL-Raten-Steuerung.

Verfahren	TYP 1 Antwortzeit [ms]	TaktTime [%]	TYP2 Antwortzeit [ms]	TaktTime [%]
SMPL (global)	93	73	763	84
GMPL				
Antwortzeit I	92	73	758	85
Antwortzeit II (WKM-Rate 1.5)	95	72	764	84
CNFL	89	74	754	86
WKM	123	68	792	81
LMPL				
Antwortzeit	175	56	771	82
CNFL	91	73	778	82
WKM	90	74	757	85

Tabelle 6.12: Optimale Antwortzeit-Ergebnisse, GMPL-Regelung

Der optimale Wert liegt bei ihr genau bei 5 %. Die Resultate der Simulation die allein über die CNFL-Rate gesteuert werden zeigen jedoch auch wie genau die CNFL-Rate zu bestimmen ist. Zusammen mit den Ergebnissen aus Simulationen mit stabiler Last, kann man schließen, daß die statische Vorgabe einer globalen CNFL-Rate, die auch für verschiedene Last-Zusammen-setzungen gute Ergebnisse liefert, kaum möglich ist.

Verfahren	Durchschnittlicher globaler MPL Wert	lokaler MPL Typ1	lokaler MPL Typ2	Eingestellter Parameter
<i>SMPL</i>	60			
<i>GMPL</i>				
<i>Antwortzeit I</i>	59			50%
<i>Antwortzeit II</i>	58			50%
<i>CNFL</i>	75			5
<i>WKM</i>	51			1.4
<i>LMPL</i>				
<i>Antwortzeit</i>	43	21	22	50%
<i>CNFL</i>	57	35	22	8%

Tabelle 6.13: Durchschnittliche MPL-Werte und Parametereinstellungen

Die lokale MPL-Regelung liefert für die Regelung mit WKM- und CNFL-Rate gute Ergebnisse. Beide sind besser als die statische MPL-Einstellung. Lediglich die antwortzeitbezogene Erkennung schneidet für beide Typen schlechter ab. Dies kann wiederum an der Einstellung der CNFL-Rate liegen. Bei der lokalen MPL-Regelung werden die typspezifischen CNFL-Raten verwendet. Die optimale Einstellung lag bei der CNFL-Raten orientierten Steuerung bei 8 %. Die Einstellung wurde beim antwortzeitbezogenen Verfahren auf dem Wert von 5 % belassen. Dies drückt sich auch im durchschnittlichen MPL-Wert aus (vgl. Tabelle 6.7). Dieser liegt im Vergleich zu den globalen Regelungsverfahren deutlich niedriger.

6.3 Simulationen mit einer realen Last

Im diesem Abschnitt werden die Ergebnisse dargestellt, die mit einer realen Datenbanklast erzielt wurden. Die Last enthält zwölf unterschiedliche Transaktionstypen. Die Transaktionsausführungen wurden über einen Zeitraum von ca. drei Stunden aufgezeichnet. Die Transaktionen in dieser Last unterscheiden sich von denen bei den synthetischen Lasten verwendeten Transaktionstypen in zwei wesentlichen Punkten. Die realen Transaktionen haben keine fixierte Anzahl von Referenzen. Die Anzahl kann somit bei zwei Transaktionen durchaus verschieden sein, obwohl sie zur selben Transaktionsklasse gezählt werden. Der zweite wesentlicher Unterschied liegt in der schwankenden Ankunftsrate. Bei den synthetischen Lasten wurde ein stochastisches Modell verwendet, um diese Schwankungen nachzuahmen.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	Summe
τ_1	9.1	3.5	3.3		5.0	0.9	0.4	0.1				0.0		22.3
τ_2	7.5	6.9	0.4	2.6	0.0	0.5	0.8	1.0	0.3	0.2	0.0			20.3
τ_3	6.4	1.3	2.8	0.0	2.6	0.2	0.7	0.1	1.1	0.4		0.0	0.0	15.6
τ_4	0.0	3.4	0.3	6.8			0.6	0.4			0.0			11.6
τ_5	3.1	4.1	0.4		0.0		0.5	0.0						8.2
τ_6	2.4	2.5	0.6		0.7		0.9	0.3						7.4
τ_7	1.3		2.6			2.3	0.1							6.2
τ_8	0.3	2.3	0.2		0.0	0.1								2.9
τ_9	0.0	1.4	0.0				1.1							2.6
τ_{10}	0.3	0.1	0.3			1.0	0.1					0.0		1.8
τ_{11}		0.9						0.2						1.1
τ_{12}		0.1												0.1
Summe:	30.0	26.6	11.0	9.4	8.3	4.9	4.1	3.3	1.4	0.6	0.0	0.0	0.0	100.0
Part.-Größe (%)	31.3	6.3	8.3	17.8	1.0	20.8	2.6	7.3	2.6	1.3	0.8	0.0	0.0	100.0

Tabelle 7.1: Relative Referenzmatrix einer realen Last (nach [Ra93]).

In Tabelle 7.1 wird das Referenzverhalten der Transaktionsklassen zusammengefaßt. Bei den Werten handelt es sich um Mittelwerte, die aus dem Mitschnitt extrahiert worden sind. Die Transaktionsklassen sind von oben nach unten entsprechend ihres Anteils an der Summe aller Referenzen angeordnet. Der prozentuale Anteil jeder Transaktionsklasse am gesamten „Referenz“

renzaufkommen“ wird in der Summationsspalte angegeben. So werden ca. 70 % aller zu verarbeitenden Referenzen von den ersten vier Transaktionsklassen verursacht. Die Transaktionsklassen von 8 bis 12 stellen nicht mehr als ca. 8 % der gesamten Referenzanzahl dar. Die Ergebnisse für diese Klassen werden in den Simulationen lediglich ind den Gaphiken der SMPL-Verfahren mit aufgeführt. In den weiteren Simulation werden zur besseren Übersichtlichkeit nur noch die Ergebniss für die anderen Transaktionsklasse (von 1 bis 7), die insgesamt 92 % aller Referenzen darstellen, aufgeführt.

Die Anordnung der Partition wurde unter dem Gesichtspunkt der Referenzhäufigkeit vorgenommen. Die Partitionen wurden von links nach rechts in absteigender Referenzhäufigkeit numeriert. So entfallen auf die ersten drei Partition ca. 68 % aller Referenzen.

Typ:	mittl. Ankunftsrate:	Abs. Anzahl von Transaktionen	mittl. Anzahl von Referenzen pro Transaktion:	Ziel-Antwortzeit:	Priorität:
τ_1	5,3 TA/s	3023	24,4	150 ms	30
τ_2	5,5 TA/s	2282	51	300 ms	30
τ_3	1,1 TA/s	728	51,7	250 ms	30
τ_4	2,4 TA/s	1099	96	500 ms	30
τ_5	2,0 TA/s	789	18,3	250 ms	30
τ_6	1,6 TA/s	749	60,5	250 ms	30
τ_7	0,8 TA/s	605	48,3	5500 ms	30
τ_8	0,8 TA/s	316	76,0	400 ms	30
τ_9	< 0,002 TA/s	252	2,6	9990 ms	30
τ_{10}	0,1 TA/s	45	780,7	5000 ms	30
τ_{11}	< 0,002 TA/s	0	0	9990 ms	30
τ_{12}	< 0,002 TA/s	1	10.000,0	9990 ms	30

Tabelle 7.2: Lastzusammensetzung für die Trace-getriebenen Simulationsläufe.

In Tabelle 7.2 werden weitere Informationen bzgl. der Simulationsparameter gezeigt. Die Simulationsdauer betrug für die reale Last 400 s. Die Ankunftsrate der Last wurde auf 25 tps gesetzt. Die Wahl der Ankunftsrate gibt in diesem Kontext an, wie viele Transaktionen pro Simulationssekunde aus dem Mitschnitt eingelesen werden. Man kann auch von der Schrittweite der Verarbeitung pro s sprechen. Im System kommen bei der gewählten Schrittweite insgesamt 10.000 Transaktionen an. Die Simulationsparameter bzgl. der anderen Komponenten

wurden beibehalten (vgl. [Kr94]). Die Schwellwerte werden beim antwortzeitbezogenen Verfahren I auf 5 % für CNFL-Rate und für das antwortzeitbezogene Verfahren II auf 1.5 für die WKM-Rate gesetzt.

6.3.1 Bestimmung eines optimalen statischen MPL-Wertes

Die Abb. 6.34 und 6.35 zeigen den Verlauf der Antwortzeit- und der entsprechenden TainTime-Kurve. Der optimale statische MPL-Wert liegt bei dieser Konstellation bei 15. In Tabelle 7.3 sind die in den einzelnen Läufen erzielten Durchschnittswerte noch einmal einzeln spezifiziert. Die Entwicklung der Antwortzeitbestandteile der einzelnen Transaktionstypen finden sich in graphischer Form im Anhang B. Diese Einzelanalysen deuten auf einen Sperrengpaß hin. Die Einstellung des MPL-Wertes ist an dieser Stelle als sehr kritisch zu bewerten. Wie man aus Tabelle 7.3 entnehmen kann, vermindert z.B. die Erhöhung des MPL-Wertes von 10 auf 15 die Antwortzeit des Transaktionstyps 1 um ein ganze Sekunde. Eine weiter Erhöhung des MPL-Wertes auf 20 verschlechtert die Antwortzeit um 0.8 s. Die Wahl des optimalen statischen MPL-Wertes liegt im Bereich von 15. Um eine besseres Ergebnis erreichen zu können, müssen die dynamischen Verfahren mit ihren MPL-Werten im Vergleich zu den bisherigen durchgeführten Simulationen in einem sehr beschränkten Intervall bleiben .

Für den MPL-Wert von 25 ergeben sich bereits Einschränkungen. Bei einer Einstellung dieses Wertes treten Durchsatzproblem auf. Dies bedeutet, daß einige Transaktionen nicht innerhalb des Simulationszeitraums beendet werden können. Dadurch ergibt sich auch ein leicht verfälschtes Bild beim Verlauf der Antwortzeit-Kurve.

SMPL	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
5	10229	9956	7298	7850	4086	8175	8619
7	2476	2560	1758	2550	1273	2264	2094
10	1424	1555	1082	1652	1162	1292	984
15	412	440	296	710	317	597	245
20	1291	1562	843	1758	1084	1269	645
25	1311	1390	970	1691	1016	1264	736

Tabelle 7.3: Antwortzeitergebnisse, statische MPL-Einstellung

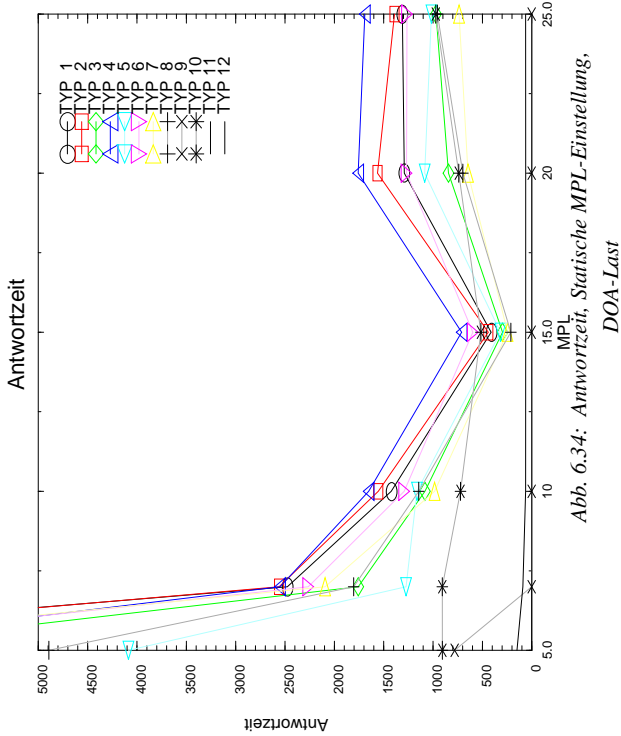


Abb. 6.34: Antwortzeit, Statische MPL-Einstellung,

DOA-Last

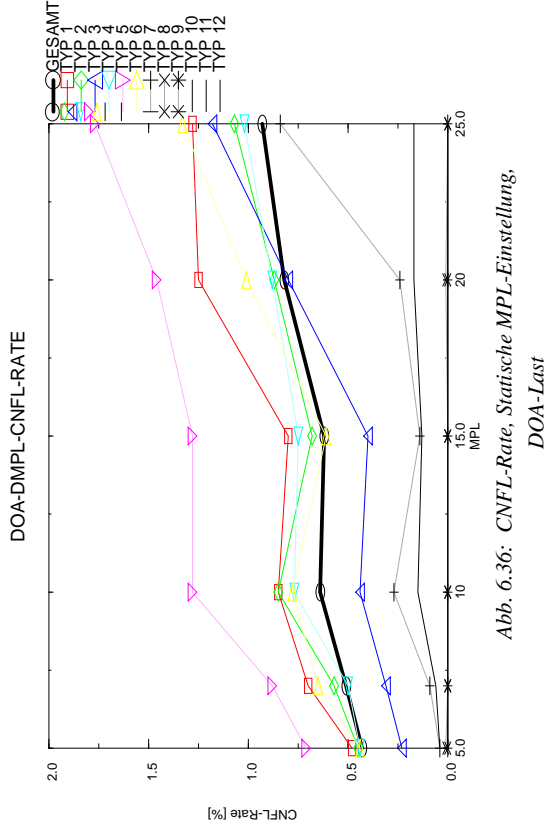


Abb. 6.36: CNFL-Rate, Statische MPL-Einstellung,

DOA-Last

Die in Abb. 6.36 gezeigten Werte für die CNFL-Rate liegen für alle MPL-Einstellungen auf einem niedrigen Niveau. So verändert sich die globale CNFL-Rate lediglich von 0.4 % auf 0.9 %. Die höchste Einzel-CNFL-Rate liegt für den Typ 5 bei 1.75 %. Diese Werte lassen sich aber nur bedingt mit den Werten aus den anderen Abschnitten vergleichen, da dort ein gleichförmigeres Lastprofil vorliegt. In der DOA-Last sind sowohl die Ankunftsrate als auch die Anzahl der Referenzen pro Transaktionstyp variabel gehalten.

Die durchschnittlichen WKM-Raten gibt Abb. 6.37 wieder. Die Werte liegen für den besten statischen Fall bei 1.3. Dieser Wert wird im Simulationslauf mit statischer MPL-Einstellung von 20 nur leicht überschritten, obwohl die Antwortzeitergebnisse stark divergieren.

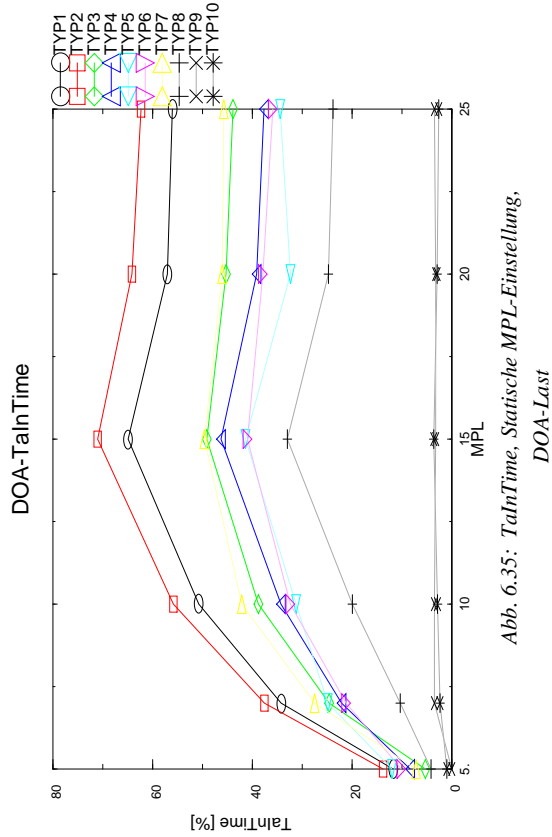


Abb. 6.35: TainTime, Statische MPL-Einstellung,

DOA-Last

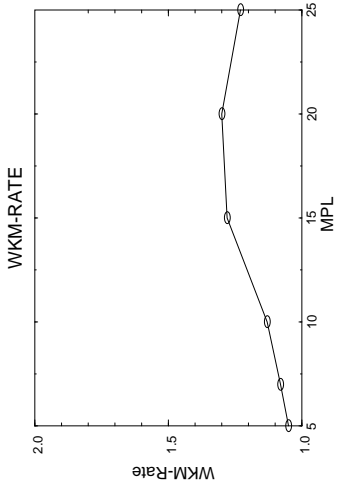


Abb. 6.37: WKM-Rate, Statische MPL-Einstellung, DOA-Last

In Abb. 6.38 werden der Verlauf der WKM-Rate, sowie der globalen CNFL-Rate in einem Simulationslauf gezeigt. Der statische MPL-Wert beträgt 20.

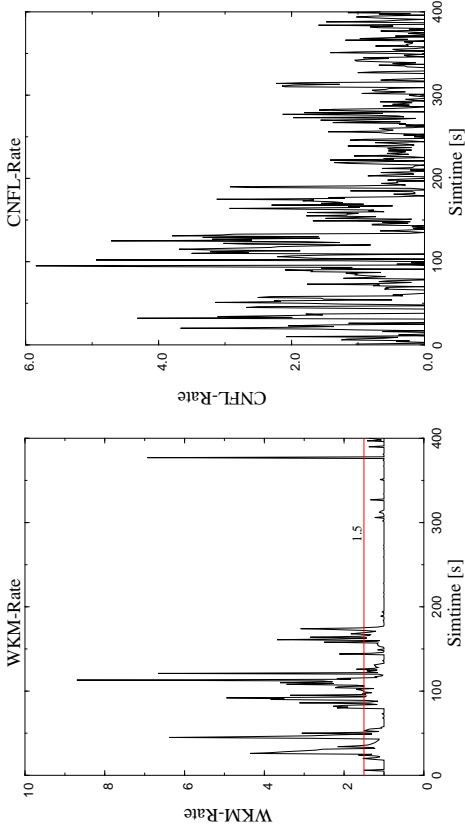


Abb. 6.38: WKM-Rate, CNFL-Rate, Statische MPL-Einstellung 20 DOA-Last

Aus den Abbildungen läßt sich entnehmen, daß vor allem in den ersten 200 s Probleme im Sperrbereich auftreten. Ein maximale WKM-Rate von fast 9 unterstreicht dies. Der CNFL-Raten-Verlauf läßt keine eindeutigen Schlüsse zu. Bei den synthetischen Lasten wurden opti-

male CNFL-Raten von 5 % bis 8 % ermittelt. Die globale CNFL-Rate erreicht diesen Wertebereich hier jedoch kaum. Dies steht in klarer Diskrepanz zu dem sehr hohen WKM-Wert. In den Simulationen mit synthetischer Last konnte zudem immer eine gewisse Verbindung zwischen CNFL-Rate und WKM-Rate konstruiert werden. In allen Fällen waren sie positiv korreliert. Diese Verbindung kann in den Grafiken auch hier abgelesen werden, der „Ausschlag“ der CNFL-Rate fällt hier jedoch sehr schwach aus.

6.3.2 Ergebnisse der globalen MPL-Regelung

Aufgrund der grossen Anzahl der Transaktionsklassen wurde von einer graphischen Darstellung der Ergebnisse Abstand genommen. Die Ergebnisse werden in tabellarischer Form dargestellt.

In den Tabellen 7.9 bis 7.8 sind die Antwortzeitergebnisse für die drei Simulationsreihen mit antwortzeitbezogener Erkennung, CNFL- und WKM-Raten festgehalten. Die Ergebnisse erreichen für **keine** der betrachteten Werte die der statischen MPL-Einstellung. Die besten Antwortzeitergebnisse aus den drei Läufen sind in Tabelle 7.9 zusammengestellt. Die zu diesen Einstellungen gehörenden MPL-Entwicklungen sind in Anhang B dargestellt.

Verfahren	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
Statischer MPL-Wert	412	440	296	710	317	597	245
Antwortzeit - Anteil I	515	619	430	930	369	634	406
Antwortzeit - Anteil II	433	462	324	679	293	571	277
CNFL-Rate	1156	1305	811	1399	1010	1087	581
WKM-Rate	562	610	456	778	456	599	389

Tabelle 7.4: Optimale Antwortzeitergebnisse, globale MPL-Regelung, DOA

Die besten Antwortzeitergebnisse bei den globalen MPL-Regelungsverfahren liefert das antwortzeitbezogene Verfahren II. An zweiter Stelle steht das antwortzeitbezogenen Verfahren I. Die Antwortzeitergebnisse sind dennoch in fast allen Fällen um 10-30 ms bzw. mindestens 100 ms (Verfahren I) schlechter als beim besten statischen Verfahren. Der durchschnittliche MPL-Wert liegt für den Fall mit 50% Antwortzeitanteil bei 11. Der MPL-Verlauf variiert beim

Verfahren I aber zwischen einem Minimum von 3 und einem Maximum von 30. Ein so breiter Bereich deutet auf Probleme bei der Stabilität des Verfahrens hin. Beim Verfahren II sind die Schwingungen auf eine geringere Bandbreite beschränkt, da die Verwendung der WKM-Rate eine zu starke Erhöhung des MPL-Wertes verhindert.

AWZ-Anteil	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
0.050	918	965	766	1020	580	888	472
0.100	1661	1706	1182	2070	865	1393	804
0.150	927	1034	671	1123	589	845	555
0.300	531	648	502	987	421	626	490
0.500	515	619	430	930	369	634	406

Tabelle 7.5: Antwortzeitanteils-Variation I, globale MPL-Regelung, DOA

AWZ-Anteil	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
0.050	1075	1145	833	1449	724	1003	543
0.100	2180	2068	1321	2615	1179	1930	1025
0.150	563	627	427	737	388	634	297
0.200	532	619	404	804	376	672	256
0.300	465	517	378	712	359	602	356
0.500	433	462	324	679	293	571	277

Tabelle 7.6: Antwortzeitanteils-Variation II, globale MPL-Regelung, DOA

CNFL-Rate	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
0.005	1411	1459	941	1666	811	1279	916
0.010	1553	1606	1022	1821	901	1417	966
0.015	1411	1459	941	1666	811	1279	916
0.020	1156	1305	811	1399	1010	1087	581
0.050	3560	4126	2095	2709	3353	3180	2274
0.100	2041	2522	1213	2727	2281	1922	1292

Tabelle 7.7: CNFL-Raten-Variation , globale MPL-Regelung, DOA

WKM-Rate	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
1.100	1447	1512	1006	1735	1030	1328	483
1.200	562	610	456	778	456	599	389
1.300	812	875	629	1102	657	843	579
1.400	1130	1265	601	1315	1056	1135	533
1.800	947	1155	481	1142	937	1198	376
2.000	1191	1326	646	1464	1193	1382	605

Tabelle 7.8: WKM-Raten-Variation , globale MPL-Regelung, DOA

Für die Steuerung mit Hilfe der CNFL-Rate ergibt sich ein anderes Bild. Die Ergebnisse sind ab einer CNFL-Rate von 5% nicht mehr relevant, da die MPL-Entwicklung nicht durch den Algorithmus sondern nur durch eine extern vorgebene Grenze (der maximal erlaubte MPL-Wert liegt bei 450) beschränkt wird. Der durchschnittliche MPL-Wert für die Simulation mit 2% Einstellung liegt bei 14.

Die globale MPL-Regelung mit WKM-Rate ist das einzige auslastungsbezogene Verfahren, bei dem die Antwortzeiten ungefähr denen entsprechen, die mit den besten statischen MPL-Einstellungen erzielt werden. Der durchschnittliche MPL-Wert liegt beim optimalen Wert der WKM-Rate von 1.2 bei ca. 14. Dies ist exakt der gleiche Wert, wie beim auslastungsbezogenen Verfahren mit CNFL-Raten-Steuerung. Wie man aus dem MPL-Verlauf entnehmen kann, senken und erhöhen die Verfahren zu unterschiedlichen Zeitpunkten den MPL-Wert (vgl. Anhang B). Betrachtet man die Antwortzeitergebnisse und die Abb. 6.38 mit dem Verlauf der WKM-Rate und CNFL-Rate, so muß man davon ausgehen, daß die WKM-Raten-Steuerung in diesem Falle die einzige Größe ist, bei der Situation und Aktion gut aufeinander abgestimmt sind.

Bei den Simulationen trat ein weiteres Problem auf. Die Rücksetzungen von Transaktionen aufgrund von Deadlock-Erkennungen wurden so zahlreich, daß Auswirkungen auf die Antwortzeit bemerkbar wurden. Dies trifft vor allem für die ersten vier Transaktionstypen zu. Im Anhang B finden sich Statistiken für drei ausgesuchte Fälle mit statischem MPL-Wert, globaler und lokaler Regelung. Es fällt auf, daß die statische MPL-Einstellung wesentlich weniger

Rücksetzungen als die beiden ausgewählten dynamischen Verfahren benötigt. Dies ist vor allem in Hinblick darauf relevant, da es sich bei den ausgewählten Fällen um die mit den besten Antwortzeit-Ergebnissen handelt.

Die Ergebnisse der Regelung des globalen MPL-Wertes bei der DOA-Last lassen sich somit zusammenfassen :

- die antwortzeitbezogene Erkennung liefert ein akzeptables Ergebnis. Die Ergebnisse aus den Simulationen mit synthetischer Last können für die CNFL-Rate nicht übernommen werden. Die Einstellung der CNFL-Rate (für die MPL-Erhöhung) von 5 % war zu hoch angesetzt worden. Die modifizierte Variante (Antwortzeit II) liefert das **beste Ergebnis** aller hier betrachteten globalen MPL-Regelungsverfahren. Die Antwortzeiten liegen für die ersten vier Transaktionstypen lediglich 20 bis 35 ms über den besten statischen Wert.
- die CNFL-Raten-bezogene Erkennung liefert keine guten Resultate. Die Verwendung der CNFL-Rate als globaler statischer Parameter erscheint nicht empfehlenswert.
- die Erkennung allein über die WKM-Rate liefert ein relativ homogenes Ergebnis in Bezug auf die MPL-Entwicklung. Die Antwortzeit-Ergebnisse sind etwas schlechter als beider antwortzeitbezogenen Erkennung I. Je nach Transaktionsklasse liegen die Antwortzeiten etwas niedriger oder höher als beim antwortzeitbezogenen Verfahren.
- alle Regelungen führen zu hohen Rücksetzquoten. Die Wiederholung bzw. der Neustart von Transaktionen wirkt sich negativ auf die durchschnittliche Antwortzeit der betroffenen Transaktionsklasse aus. Unter Berücksichtigung der MPL-Kurvenläufe (im Anhang) muß somit festgestellt werden, daß die MPL-Werte in einem zu großen Intervall schwanken.

6.3.3 Ergebnisse der lokalen MPL-Regelung

In den Tabelle 7.11 bis 7.12 werden die Ergebnisse für die lokale MPL-Regelung dargestellt. Bei keiner der durchgeführten Simulationen konnte ein für alle Transaktionsklassen gutes Antwortzeitergebnis erzielt werden. So lagen z.B. die besten Antwortzeiten bei der antwortzeitbezogenen Erkennung und der CNFL-Raten-Steuerung für den Transaktionstypen 1 bei ca. 1 s. Die beste Antwortzeit lag bei Verwendung der WKM-Rate bei unakzeptablen 9 s. In Tabelle 7.9 sind die besten erzielten Antwortzeitergebnisse noch einmal zusammengefaßt. Die Situation ergibt kein einheitliches Bild. Einige der Transaktionen, wie z.B. der Transaktionstyp 3, werden bei der lokalen MPL-Einstellung bevorzugt. Vor allem die Transaktionen mit einer hohen Ankunftsrate wie der Typ 1 brauchen jedoch wesentlich mehr Zeit zur Verarbeitung.

Verfahren	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
Statischer MPL-Wert	412	440	296	710	317	597	245
Antwortzeit - Anteil	1026	1043	459	2141	742	872	268
CNFL-Rate	1067	1060	227	1249	722	683	236

Tabelle 7.9: Optimale Antwortzeitergebnisse, lokale MPL-Regelung, DOA

Bei den Verfahren konnten keine uneingeschränkter MPL-Anstiege beobachtet werden. Als generelle Tendenz kann jedoch festgehalten werden, daß die durchschnittliche Summe der lokalen MPL-Werte bei der lokalen MPL-Steuerung höher als die MPL-Werte bei der globalen Regelung ist.

In der Tabelle 7.9 sind die durchschnittlichen MPL-Werte für die Parameter, mit denen die besten mit der lokalen Regelung erzielbaren Antwortzeiten erreicht wurden, dokumentiert.

Verfahren	τ_1 MPL	τ_2 MPL	τ_3 MPL	τ_4 MPL	τ_5 MPL	τ_6 MPL	τ_7 MPL
Antwortzeit-Anteil	3	4	4	5	3	3	5
CNFL-Rate	3	5	3	4	3	4	3
WKM-Rate	18	5	2	2	2	2	2

Tabelle 7.10: Durchschnittliche MPL-Werte, lokale MPL-Regelung, DOA

7. Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein neues Konzept für die Lastregelung in zentralisierten Transaktionsystemen untersucht. Grundlage des Konzeptes ist die automatische Einhaltung von extern vorgegebenen Antwortzeitzielen für einzelne Transaktionsklassen. Bei Verfehlung der Antwortzeitziele unternimmt das System Regelungsmaßnahmen, um eine bestmögliche Zielerreichung zu gewährleisten. Als „Steuerungsinstrumente“ wurden die Verwendung des MPL-Wertes und der Einsatz von Prioritäten untersucht. Die MPL-Verfahren lassen sich in die Steuerung mit globalen und lokalen MPL-Werten unterteilen. Beim globalen Verfahren wurde nur der über alle Transaktionsklassen gültige globale MPL-Wert verändert. Bei der lokalen Regelung wurde zusätzlich noch vor der eigentlichen Regelung untersucht, ob der Engpaßwert oder die Engpässe für die Mehrzahl oder nur für eine Minderheit der Transaktionsklassen vorhanden ist. Je nach dem Resultat wird dann entweder eine Regelung des globalen MPL-Wertes oder eine Regelung des transaktionsklassenspezifischen lokalen MPL-Wertes vorgenommen. Der Einsatz von Prioritäten wurde nur beim CPU-Engpaß untersucht. Gegenstand der Analyse waren die Varianten Earliest-Deadline und Performance-Index.

In der ersten Simulationsreihe wurde die **CPU-Engpaß-Behandlung** untersucht. Es konnten festgestellt werden, daß für die betrachteten Fälle eine CPU-WAIT-Anteil von 25% guten Resultate erzielte. Die Einstellung des BOT-WS-Anteils war nur sekundär relevant. Es konnten bestimmt werden, daß der CPU-Auslastungsindikator relativ nahe an der maximalen Auslastungsgrenze orientiert werden sollte. Die Ergebnisse für die antwortzeitbezogene und eine auslastungsbezogene Erkennung zeigten, daß beide Varianten verwendbar sind.

Die Experimente mit lokaler MPL-Regelung und antwortzeitbezogener Erkennung deckten eine Problem bei der Engpaß-Erkennung auf. Die lokale MPL-Regelung reduziert die MPL-Werte derjenigen Transaktionsklassen, die unter dem CPU-Engpaß am meisten leiden (in Form der Antwortzeit). Transaktionsklassen, die auch zum CPU-Engpaß beitragen, werden aufgrund ihrer Antwortzeit-Zusammensetzung nicht „zur Rechenschaft“ gezogen. Für diese Transaktionsklassen (in den Simulationen ist es der lange Transaktionstyp) wird ein CPU-Engpaß nicht erkannt und folglich keine MPL-Reduktion vorgenommen. Ursache dieses Verhaltens ist die globale Vorgabe von Antwortzeit-Prozentanteilen, die keine Rücksicht auf die transaktionsklassenspezifischen Antwortzeit-Zusammensetzungen nimmt. Die Vorgabe von individuellen Vorgabevektoren (vgl. Abb. 4.4) macht jedoch einen manuellen bzw. externen

AWZ-Anteil	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
0.050	3008	16576	1892	6124	7389	5166	1558
0.100	1026	1043	459	2141	742	872	268
0.150	10673	1060	227	1249	722	683	236
0.200	17881	9923	8452	7252	3981	5464	3304
0.300	8550	10334	348	18735	14225	7439	8636
0.500	923	14240	798	8681	4604	8034	1710

Tabelle 7.11: Antwortzeitanteils-Variation , lokale MPL-Regelung, DOA

CNFL-Rate	τ_1 [ms]	τ_2 [ms]	τ_3 [ms]	τ_4 [ms]	τ_5 [ms]	τ_6 [ms]	τ_7 [ms]
0.050	30080	16576	1892	6124	7389	5166	1558
0.100	1026	1043	457	2141	742	872	268
0.150	1067	1060	227	1249	722	683	236
0.200	17881	9923	8452	7252	3981	5464	3304
0.300	8550	10334	348	18735	14225	7439	8636
0.500	923	14240	798	8681	4604	8034	1710

Tabelle 7.12: CNFL-Raten-Variation, lokale MPL-Regelung, DOA

Bei den durchgeführten Simulationen lag die Rücksetzquote etwas höher als bei den globalen MPL-Verfahren. Dies betraf vor allem den Transaktionstyp 4. Als Ergebnis der lokalen MPL-Regelung kann festgehalten werden, daß das Verfahren bei der DOA-Last zu inakzeptablen Resultaten führt. Die Gründe für dieses Versagen, traten bereits im 2-Typen Fall mit synthetischer Lastgenerierung auf. Zum einem werden die Engpässe für einzelne Typen aufgrund global vorgegebener Antwortzeit-Prozentsätze nicht adäquat erkannt. Zum anderen macht sich auch bei der MPL-Erhöhung die starke Abhängigkeit von der Laststruktur bemerkbar. Die Verwendung der CNFL-Rate als Sperrengpassindikator kann nicht für alle Typen mit dem gleichen Wert erfolgen.

Eingriff eines Experten notwendig (falls die Modifikation der Anteile nicht auch automatisiert werden kann). Als elegantere Methode bietet sich die Einbindung der `TaInTime`-Anteile bei der Analyse an. Die isolierte Erkennung eines CPU-Engpasses für eine Transaktionsklasse und eine damit generell verbundene MPL-Reduktion, sollte nur unter Einbeziehung der anderen Transaktionsklassen geschehen. Die lokale MPL-Reduktion sollte nur dann durchgeführt werden, wenn die jeweilige Transaktionsklasse i alleiniger Verursacher des Engpasses ist oder bei mehreren Verursachern, die `TaInTime`-Anteile der betrachteten Klasse i die höchsten unter allen Verursachern ist. Falls dies für Klasse i nicht gilt, so wird der MPL-Wert der verursachenden Klasse mit den höchsten `TaInTime`-Anteilen verringert.

Die Versuche, den Verarbeitungsmix durch die Verwendung von Prioritäten besser zu gestalten, haben gezeigt, daß ein einfacher Einsatz von verschiedenen Prioritätsalgorithmen zu keiner allgemeinen Verbesserung der Resultate führen. Beim `SMPL`-Verfahren wurden beim `ED`-Verfahren die Ergebnisse für den kurzen Transaktionstyp auf Kosten des langen Transaktionstyps verbessert. Das PI lieferte in der verwendeten Form die schlechtesten Resultate.

Bei `GMPL`-Regelung konnte ein ähnliches Verhalten beobachtet werden. Beide Prioritätsalgorithmen führen bei Verwendung auf `BOT-WS`- und `CPU`-Ebene zu einer Verringerung der `TaInTime`-Anteile für die beiden Typen. In den Simulationen zeigte sich, daß ein Einsatz der Prioritätsalgorithmen sich auf die `BOT-WS` beschränken sollte. Der Einsatz von `ED` auf der `BOT`-Ebene ergab für die `GMPL`-Regelung die ausgewogensten `TaInTime`-Verteilung zwischen den beiden Transaktionstypen. Im besten beobachteten Fall lagen die `TaInTime`-Anteile bei dieser Einstellung über bzw. mindestens in gleicher Höhe der mit den besten statischen und dynamischen Verfahren (`GMPL`-Regelung mit `FCFS`) erzielten Ergebnisse. Die Regelung unter Verwendung des PI zeigte ein sehr instabiles Verhalten der Resultate in Bezug auf den `MPL[0]`-Wert. Sowohl die Antwortzeiten als auch die `TaInTime`-Anteil liegen über denen der anderen Verfahren. In der augenblicklichen Form ist von der Verwendung des PI aufgrund seines stark schwingenden Verhaltens abzusehen. Als Verbesserungsmöglichkeit bietet sich die Kontingentierung der `MPL`-Slots (vgl. Abschnitt 5.5.3) an.

Der zweite Hauptkomplex der Untersuchung stellt die **Behandlung von Sperrengpässen** dar. In den Simulationen wurden drei verschiedene Größen zur Erkennung verwendet. Es läßt sich festhalten, daß der originäre antwortzeitbezogene Ansatz gute Ergebnisse liefern kann. Die

Prozentsätze für die Sperrbearbeitung sollten relativ hoch angesetzt werden. In den Simulationen ergaben sich Werte von 50%. Problematisch bleibt jedoch die für alle Transaktionsklassen verbindliche Vorgabe von Antwortzeit-Prozentsätzen. Diese allgemeine Vorgabe wirkte sich bei den beiden Hauptstrategien der globalen und lokalen `MPL`-Regelung unterschiedlich aus. Bei der globalen `MPL`-Regelung ist die Aggregation der verschiedenen Transaktionsklassen das Problem. Bei der lokalen `MPL`-Regelung werden diese generellen Prozentsätze unabhängig von der Kenntnis der Antwortzeitstruktur der betrachteten Transaktionsklassen vorgenommen.

Als möglicher Lösungsansatz bietet sich, wie auch bei der `CPU`-Engpaß-Erkennung, die Vorgabe von transaktionsklassenspezifischen Antwortzeitanteilen an. Ein anderen Ansatz kombiniert allgemein vorgegebene Antwortzeitanteile mit lokalen Beobachtungen. Dies macht es aber notwendig ein gedächtnisorientiertes Verfahren zu realisieren. Aufgabe dieser Komponente ist, die Antwortzeit-Zusammensetzung jeder einzelnen Transaktionsklassen zu überwaachen und aus den Werten Zu- oder Abschläge für die globalen Antwortzeitanteile zu berechnen. Die Bestimmung des Engpasses erfolgt dann unter Berücksichtigung der globalen Prozentsätze und den spezifischen Zu- und Abschläge.

In den Simulationen zeigte sich auch, daß Regelungen mit den Parametern `CNFL`-Rate und `WKM`-Rate punktuell gute Ergebnisse erzielen können. Es lassen sich aber keine allgemeingültigen Heuristiken oder Grenzwerte ("Datenbank-Naturkonstanten") ableiten. Die Verwendung dieser Größen bietet sich vor allem bei der `MPL`-Erhöhung an. Denkbar ist auch eine Kombination der beiden Werte. Der Einsatz der `CNFL`-Rate ist ohne Einbeziehung der Kenntnisse der Transaktionscharakteristiken nicht empfehlenswert. Ein Einsatz dieser Größe bietet sich nur bei Regelung des lokalen `MPL`-Wertes an, da die Abhängigkeit der durchschnittlichen Antwortzeit von der `CNFL`-Rate für lange und kurze Transaktionstypen sehr unterschiedlich ist. Die Vorgabe von transaktionsklassenspezifischen `CNFL`-Raten kann unter Umständen auch durch ein gedächtnisorientiertes Verfahren realisiert werden. Die Ergebnisse mit der `WKM`-Rate als alleiniger Steuerungsparameter oder in Kombination mit dem antwortzeitbezogenen Verfahren (Antwortzeit-Verfahren II) zeigen, daß die `WKM`-Rate wesentlich stabiler in Bezug auf die Laststruktur (Art der Transaktionstypen und Zusammensetzung der Gesamlast) ist als die `CNFL`-Rate. Vor allem die Simulationsergebnisse mit der `DOA`-Last, die aus sehr verschiedenen Transaktionstypen zusammengesetzt ist, zeigen, daß bei globaler Regelung der Einsatz der `WKM`-Rate beim Test vor einer `MPL`-Erhöhung die besten Ergebnisse liefert.

(vgl. Abschnitt 6.4.2). Diese alleinige Verwendung der WKM-Rate bietet sich primär für die globale MPL-Regelung an. Die Verwendung der WKM-Rate beim LMPL-Verfahren stößt an Grenzen, da die WKM-Rate keine transaktionsklassenspezifischen Größe ist. Ob der Einsatz einer globalen Größe bei einem lokalen Verfahren zu akzeptablen Ergebnissen führt kann nur simulativ beantwortet werden. Aus theoretischer bzw. konzeptioneller Sicht ist ein solcher Ansatz schon vorab kritisch zu beurteilen. Für das LMPL-Verfahren ist auch eine Kombination der beiden Metriken denkbar. So sollte der Test vor einer MPL-Erhöhung unter Einbeziehung beider Werte vollzogen werden. Zusammen mit einer durchgängigen Einbeziehung der lokalen MPL-Werte bei MPL-Erhöhungsentscheidungen sind wesentliche Verbesserungen denkbar.

Die mit den synthetischen Lasten durchgeführten Simulationen arbeiten ohne Ausnahme mit Transaktionstypen, die eine feste Anzahl von Referenzen durchführen. Es empfiehlt sich die Simulationen auch mit Transaktionstypen durchzuführen, deren Referenzanzahl in einem bestimmten Spektrum variiert.

8. Anhang

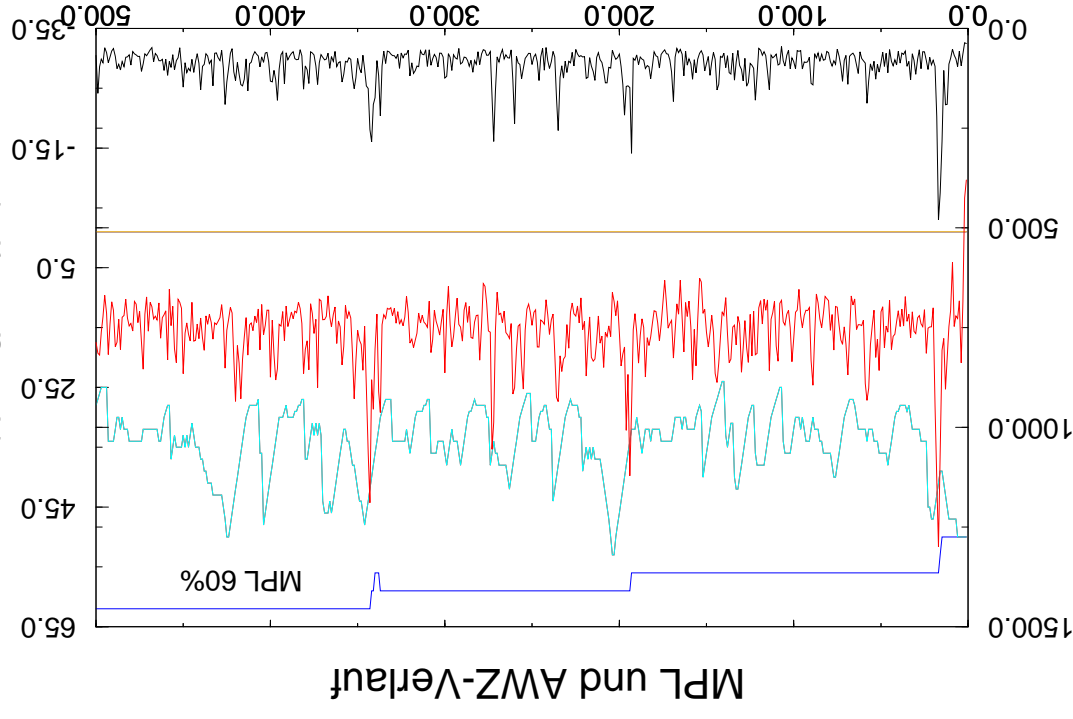
8.1 Literaturverzeichnis

- [AG89] Abbott, Robert, Garcia-Molina, Hector: *Scheduling Real-Time Transactions with Disk-Resident Data*. Proc. 15th Int. Conf. on Very Large Data Bases, 385-396, 1989.
- [BH87] Bernstein, P.A., Hadzilacos, V., Goodman, N.: *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, Reading, 1987.
- [BCL93] Brown, K.P., Carey J.C., Livny M.: *Towards an Autopilot in the DBMS Performance Cockpit*. International Conference on Very Large Databases, Dublin, 1993.
- [CKL89] Carey, M. J., Krishnamurthi, S., Livny, M.: *Load-Control for Locking: The 'Half-and-Half' Approach*, Computer Sciences Dept., University of Wisconsin, 1989.
- [EN89] Elmastri, R., Navathe, S.B.: *Fundamentals of Database Systems*. Benjamin Cummings Publishing Company, Redwood City, Ca, 1989.
- [EV93] Evans, J.B.: *Structures of Discrete Event Simulation*. Ellis Horwood Verlag, New York/Chichester, Brisbane, Toronto, 1993.
- [FGN93] Fergusson, D., Georgiadis, L., Nikolaou, C., Davies, K.: *Satisfying Response Time Goals in Transaction Processing Systems*, 2nd International Conference on Parallel and Distributed Information Systems, San Diego, 1993.
- [FNG93] Fergusson, D., Nikolaou, C., Georgiadis, L.: *Goal Oriented, Adaptive Transaction Routing for High Performance Transaction Processing Systems*, 2nd International Conference on Parallel and Distributed Information Systems, San Diego, 1993.
- [GR93] Gray, J., Reuter, A.: *Transaction Processing : Concepts and Techniques*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [Hä+87] Härder, T. u.a.: *Datenbank-Handbuch*. Hrsg: Lockemann, P.C., Schmidt, J.W., Springer-Verlag Berlin, Heidelberg, 1987.
- [HLC91] Haritsa, J.R., Livny, M., Carey, J.C.: *Earliest Deadline Scheduling for Real-Time Database Systems*, Proc. of the 12th Real Time Systems Symposium, 1991.
- [Kr94] Krink, G.S.: *Simulation von adaptiven Lastkontrollverfahren zur Transaktionsverarbeitung* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, April 1994.
- [Li89a] Livny, M.: *DeNet Overview*. Computer Science Dept., University of Madison-Wisconsin 1989.
- [Li89b] Livny, M.: *DeNet User's Guide Ver. 1.5*. University of Wisconsin-Madison, Nov. 1989.
- [Ma91] Marek, R.: *Simulation eines Shared-Nothing-Systems zur parallelen Query-Verarbeitung* Technischer Bericht, Fachbereich Informatik, Universität Kaiserslautern, 1991.
- [MW90] Mönkeberg, A., Weikum, G.: *Conflict-driven Load Control for the Avoidance of Data-Contention thrashing*. Database Research Group, ETH Zürich, 1990.
- [MW92] Mönkeberg, A., Weikum, G.: *Performance Evaluation of an Adaptive Load Control Method for the Avoidance of Data-Contention Thrashing*. Proc. of the 18th VLDB-Conference, Vancouver, British Columbia, Canada, 1992.
- [PLC92] Pang, H., Livny, M., Carey, J.C.: *Transaction Scheduling in Multiclass Real-Time Database Systems*, Proc. of the 13th Real Time Systems Symposium, 1992
- [RFL89] Rahm, E., Fergusson, D., Leonidas, G. u.a.: *Goal-oriented Workload Management in Locally Distributed Transaction Systems*. IBM Research Report RC 14712, IBM T.J. Watson Research Center, Yorktown Heights, 1989.

- [ra93] Rahm, E.: *Hochleistungstransaktionssysteme*. Vieweg-Verlag Braunschweig/Wiesbaden, 1993.
- [ra94] Rahm, E.: *Mehrrechner-Datenbanksysteme*. Addison-Wesley Verlag Bonn/Paris, 1994.
- [spg91] Silberschatz, A., Peterson, J., Galvin, P.: *Operating System Concepts*. Addison-Wesley Publishing New York, 1991.
- [st91] Stöhr, T.: *Simulation eines zentralen Sperrverfahrens für nahe gekoppelte DB-Sharing-Systeme*. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1991.
- [st93] Stöhr, T.: *Simulation eines Shared-Disk-Systems zur parallelen Query-Bearbeitung*. Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, März 1993.
- [th92] Thomasian, A.: *Thrashing in Two-Phase Locking Revisited*. Proc. of the 8th International Conference on Data Engineering, 1992.
- [ta85] Tay, Y., C.: *Locking Performance in Centralized Databases*. Proceedings of the ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, Waterloo, Canada, April 1984.
- [vo93] Vossen, G.: *Grundlagen der Transaktionsverarbeitung*. Addison Wesley, 1993.
- [wi83] Wirth, N.: *Programming in MODULA-2*. 2nd Ed., Springer Verlag Berlin/Heidelberg/ New York, 1983.
- [wmmz94] Weikum, G., Hasse, C., Mönkeberg, A., Zabback, P.: *The COMFORT Automatic Tuning Project*. Information Systems, No. 5, pp. 381-432, 1994.
- [wo91] Wollenhaupt, G.: *Simulation von erweiterten Speicherhierarchien zur Transaktionsverarbeitung*. Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, Mai 1991.

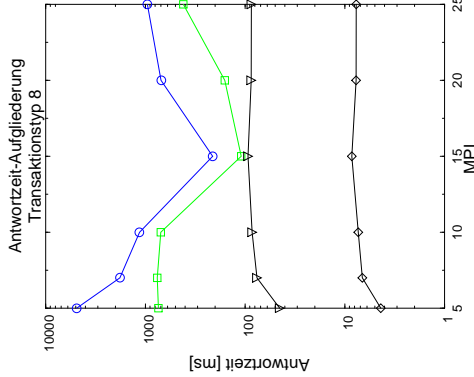
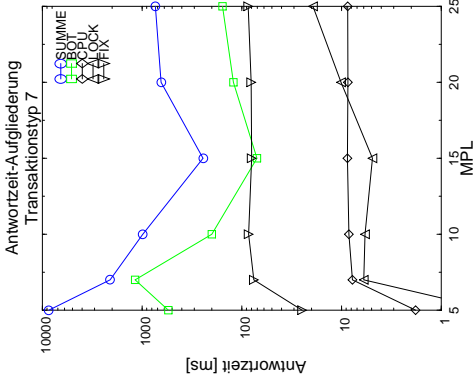
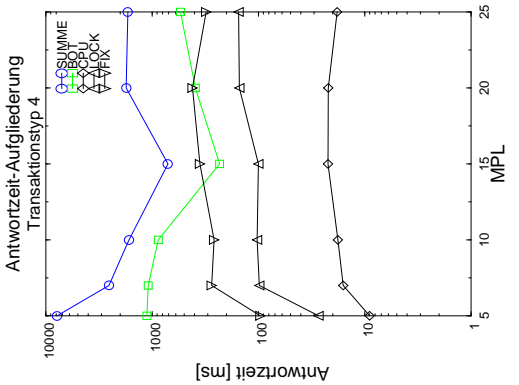
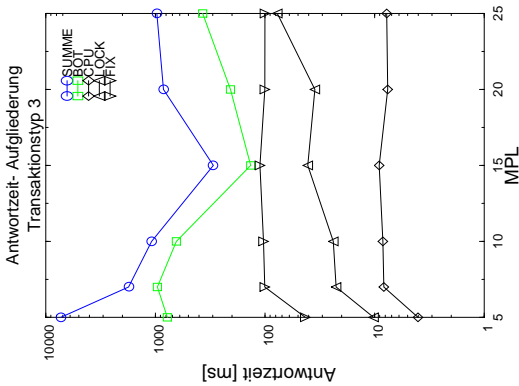
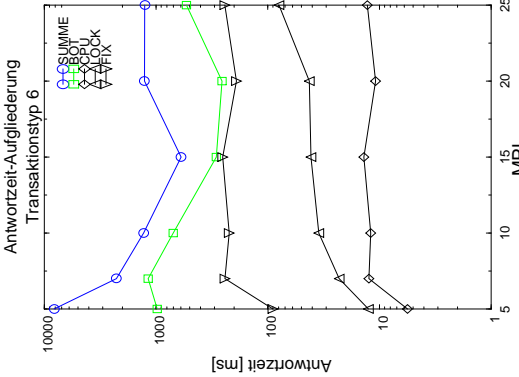
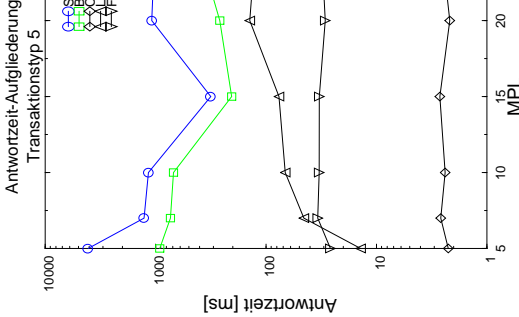
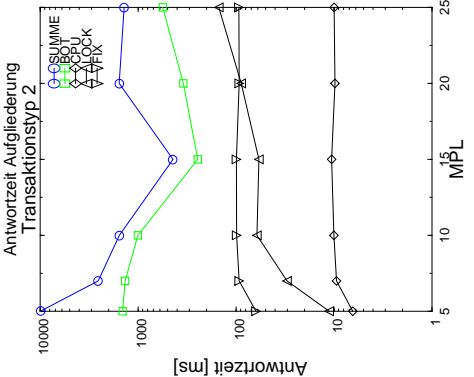
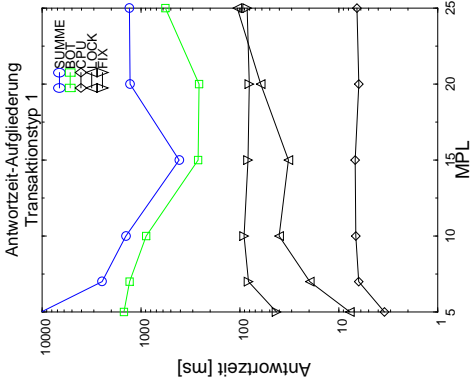
8.2 Anhang A

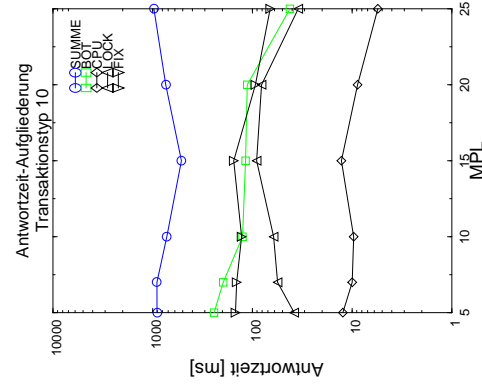
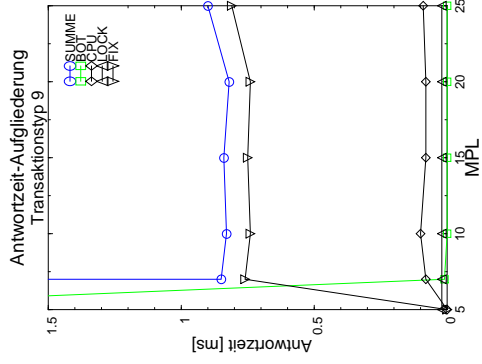
MPL und AWZ- Verlauf bei variierenden Last
(Sperrengpaß 2 Typen)



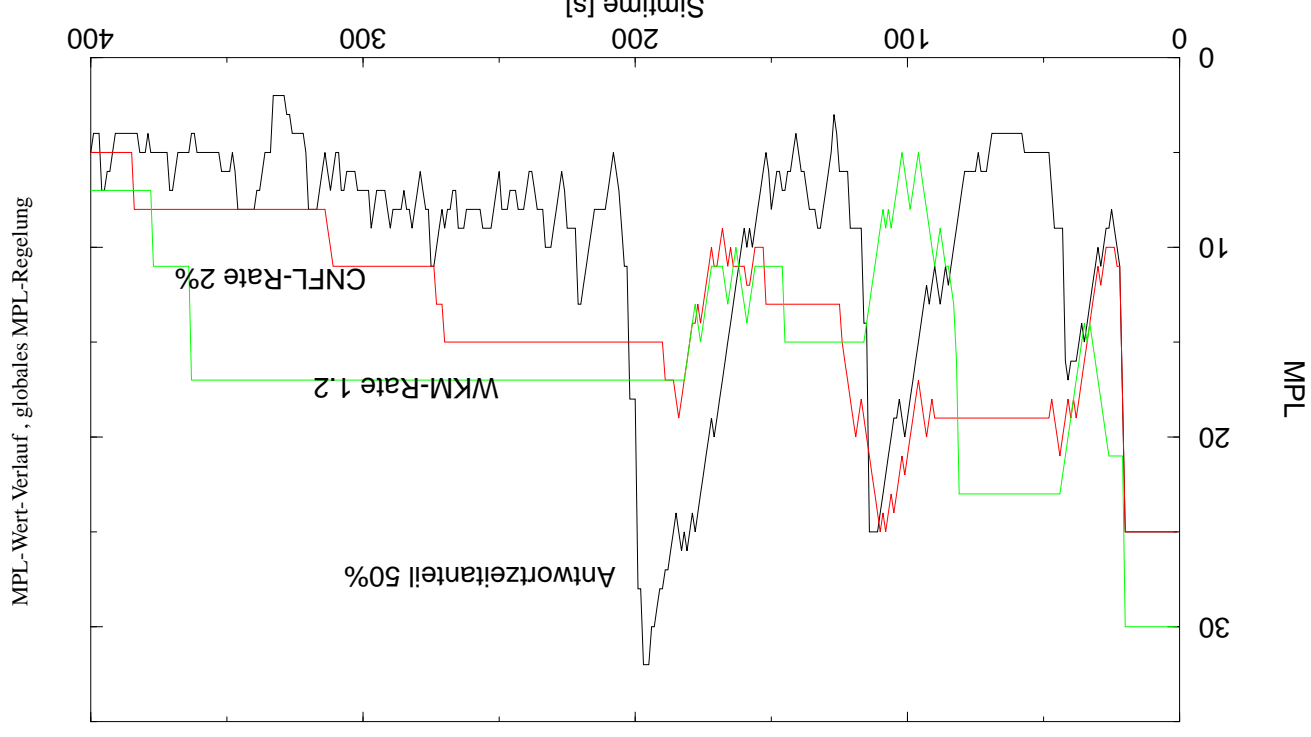
8.3 Anhang B

Antwortzeit-Aufgliederung DOA-Last





MPL Verlauf - GMPL Verfahren



Verarbeitungsstatistiken

Globaler statischer MPL-Wert 15

	Arrived:	Finished & Aborted:		Aborted:	total:		Committed:		overdue:	
		abs.	in %		abs.	in %	abs.	in %	abs.	in %
1	3023	3023	28	0.93	3023	100.00	2030	67.15	993	32.85
2	2282	2279	23	1.01	2279	100.00	1548	67.92	731	32.08
3	729	727	3	0.41	727	100.00	567	77.99	160	22.01
4	1087	1087	39	3.59	1087	100.00	811	74.61	276	25.39
5	789	789	3	0.38	789	100.00	562	71.23	227	28.77
6	750	748	24	3.21	748	100.00	460	61.50	288	38.50
7	605	605	0	0.00	605	100.00	580	95.87	25	4.13
8	316	316	0	0.00	316	100.00	232	73.42	84	26.58
9	252	252	0	0.00	252	100.00	252	100.00	0	0.00
10	45	45	15	33.33	45	100.00	26	57.78	19	42.22
11	0	0	0	0.00	0	0.00	0	0.00	0	0.00
12	1	1	0	0.00	1	100.00	0	0.00	0	100.00
0	9879	9872	135	1.37	9872	100.00	7068	71.60	2804	28.40

Globale MPL-Regelung Antwortzeitanteil 50%

	Arrived:	Finished & Aborted:		Aborted:	total:		Committed:		overdue:	
		abs.	in %		abs.	in %	abs.	in %	abs.	in %
1	3023	3023	19	0.63	3023	100.00	1692	55.97	1331	44.03
2	2282	2279	35	1.54	2279	100.00	1355	59.46	924	40.54
3	729	725	27	3.72	725	100.00	482	66.48	243	33.52
4	1087	1087	377	34.68	1087	100.00	724	66.61	363	33.39
5	789	789	41	5.20	789	100.00	498	63.12	291	36.88
6	750	747	41	5.49	747	100.00	404	54.08	343	45.92
7	605	600	15	2.50	600	100.00	562	93.67	38	6.33
8	316	316	0	0.00	316	100.00	175	55.38	141	44.62
9	252	252	0	0.00	252	100.00	252	100.00	0	0.00
10	45	45	25	55.56	45	100.00	24	53.33	21	46.67
11	0	0	0	0.00	0	0.00	0	0.00	0	0.00
12	1	1	0	0.00	1	100.00	0	0.00	1	100.00
0	9879	9864	580	5.88	9864	100.00	6165	62.50	3699	37.50

LMPL-Regelung Antwortzeitanteil 10%

	Arrived:	Finished & Aborted:		Aborted:	total:		Committed:		overdue:	
		abs.	in %		abs.	in %	abs.	in %	abs.	in %
1	3023	3011	35	1.16	3011	100.00	1632	54.20	1379	45.80
2	2282	2281	28	1.23	2281	100.00	1337	58.61	944	41.39
3	729	726	3	0.41	726	100.00	564	77.69	162	22.31
4	1087	1087	457	42.04	1087	100.00	712	65.50	375	34.50
5	789	789	6	0.76	789	100.00	517	65.53	272	34.47
6	750	748	32	4.28	748	100.00	526	70.32	222	29.68
7	605	608	0	0.00	608	100.00	575	94.57	33	5.43
8	316	316	0	0.00	316	100.00	265	83.86	51	16.14
9	252	252	0	0.00	252	100.00	252	100.00	0	0.00
10	45	45	25	55.56	45	100.00	21	46.67	24	53.33
11	0	0	0	0.00	0	0.00	0	0.00	0	0.00
12	1	1	0	0.00	1	100.00	0	0.00	1	100.00
0	9879	9861	594	6.02	9861	100.00	6311	64.00	3550	36.00